ESD TDR 65-183
ESTI FILE COPY

-65-183

ESD RECORD COPY
RETURN TO
SCIENTIFIC & TECHNICAL INFORMATION DIVISION
(ESTI), BUILDING 1211
COPY NR.________ OF________ COPIES

(FINAL REPORT)

MODELS OF COMMAND AND CONTROL SYSTEMS
(WITH APPLICATIONS TO EXERCISE AND EVALUATION)

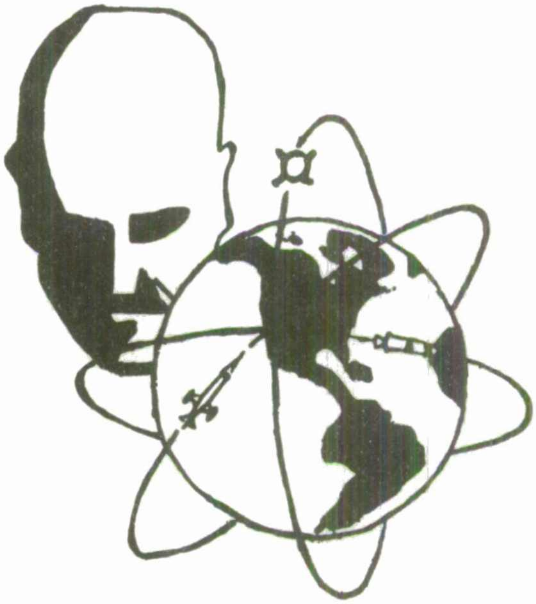
TECHNICAL DOCUMENTARY REPORT NO. ESD-TDR-65-183

FEBRUARY 1965

ESTI PROCESSED
☐ DDC TAB ☐ PROJ OFFICER
☐ ACCESSION MASTER FILE
☐ ____________
DATE____________
ESTI CONTROL NR. AL 45859
CY NR. 1 OF 1 CYS

Peter Kugel
Martin F. Owens

DIRECTORATE OF COMPUTERS
ELECTRONIC SYSTEMS DIVISION
AIR FORCE SYSTEMS COMMAND
UNITED STATES AIR FORCE
L. G. Hanscom Field, Bedford, Massachusetts



Project 2801

AD0 615549

(Prepared under Contract No. AF 19 (628)-2455 by the Technical Operations Research, Burlington, Massachusetts.)

DDC AVAILABILITY NOTICE

Copies have been deposited with the Defense Documentation Center. (DDC)

DISSEMINATION NOTICE

Copies available at the Clearing House for Federal Scientific & Technical Information. (CFSTI) (Formerly OTS)

LEGAL NOTICE

When US Government drawings, specifications or other data are used for any purpose other than a definitely related government procurement operation, the government thereby incurs no responsibility nor any obligation whatsoever; and the fact that the government may have formulated, furnished, or in any way supplied the said drawings, specifications, or other data is not to be regarded by implication or otherwise as in any manner licensing the holder or any other person or conveying any rights or permission to manufacture, use, or sell any patented invention that may in any way be related thereto.

OTHER NOTICES

Do not return this copy. Retain or destroy.

HEADQUARTERS
ELECTRONIC SYSTEMS DIVISION
AIR FORCE SYSTEMS COMMAND
UNITED STATES AIR FORCE

LAURENCE G. HANSCOM FIELD BEDFORD, MASSACHUSETTS 01731

REPLY TO ATTN OF: ESEP/65-117/5322 4 June 1965

SUBJECT: Security Review

TO: ESTI (Lt. Rives)

I can see no objection to releasing this material to OTS.

John T. O'Brien

JOHN T. O'BRIEN
Chief, Public Information Division
Information Office

1 Atch
n/c

ESD-TR-65-183

ESD-TDR-65-183

(FINAL REPORT)

MODELS OF COMMAND AND CONTROL SYSTEMS
(WITH APPLICATIONS TO EXERCISE AND EVALUATION)

TECHNICAL DOCUMENTARY REPORT NO. ESD-TDR-65-183

FEBRUARY 1965

Peter Kugel
Martin F. Owens

DIRECTORATE OF COMPUTERS
ELECTRONIC SYSTEMS DIVISION
AIR FORCE SYSTEMS COMMAND
UNITED STATES AIR FORCE
L. G. Hanscom Field, Bedford, Massachusetts

Project 2801

(Prepared under Contract No. AF 19 (628)-2455 by the Technical Operations Research, Burlington, Massachusetts.)

FOREWORD

The work described in this report was performed by Technical Operations Research for the Systems Design Laboratory of the Electronics System Division of the United States Air Force under Contract AF 19(628)-2455. The purpose of this contract was to develop techniques to improve methods used in constructing, controlling, and evaluating command and control system exercises. The work was based on an examination of records of exercises of an existing command and control system (473L). Models were developed to describe various aspects of these exercises, using existing mathematical techniques.

The contract monitors were Lt. Don Parker (ESRC), Lt. F. A. Fresh (ESRC), and Robert P. Savoy (ESRC). Martin F. Owens, Robert A. Langevin, Stanley LaVallee, and Peter Kugel (Project Leader) worked on this contract for Technical Operations Research. Department D-25 of The MITRE Corporation, particularly Mr. John Burns and Dr. John Proctor, assisted in making available exercise records and in numerous discussions; this help is gratefully acknowledged. This report describes roughly the results of the second half of the contract. The work described was performed from 30 November 1963 to 31 January 1965. The results of the first half of this contract are described in report (TO-B 63-108) entitled "Some Techniques to Help Improve Methods for Exercising and Evaluating Command and Control Systems."

ABSTRACT

Five models of the activities of command and control systems are described to provide a precise, if not necessarily quantitative, framework within which the behavior of command and control systems can be studied. Such models are intended as the basis for theories into which empirical data about this behavior, derived from the observation of exercises, could be fitted to provide predictions about the future behavior of particular systems.

The Deductive Inference Model describes information processing as the manipulation of strings according to explicitly given rules. In terms of such a description, this model deals with the processes of problem identification and problem solving.

The Inductive Inference Model deals with information processing for which the system must derive the rules that are to be used. It relates the assumptions that such a system makes and the inductive strategies that it uses to the adequacy of its predictions and generalizations.

The Value Model treats a command and control system as a system that applies the values of the commander. It attempts to relate measurable features of the values held by personnel to the kinds of decisions that they make.

The Semantic Model tries to deal with the manner in which command and control systems and their personnel represent their information about their environment.

The Finite Automaton Model treats a command and control system and exercise controllers in certain types of controlled exercises as coupled sequential machines (finite automata). It provides a vehicle for studying the ability of the exerciser to control the behavior of the system and for studying an exercise as a learning situation.

This technical documentary report has been reviewed and is approved:

Robert P. Savoy

ROBERT P. SAVOY
Task Scientist, Task 280103
Computer Division
Directorate of Computers

C. K. Wilcox, Lt Colonel, USAF
for PAUL G. GALENTINE, JR.
Colonel, USAF
Director of Computers
Deputy for Engineering & Technology

TABLE OF CONTENTS

TABLE OF CONTENTS (Cont'd.)

TABLE OF CONTENTS (Cont'd.)

TABLE OF CONTENTS (Cont'd.)

LIST OF ILLUSTRATIONS

LIST OF TABLES

MODELS OF COMMAND AND CONTROL SYSTEMS (WITH APPLICATIONS TO EXERCISE AND EVALUATION)

PROBLEMS AND SOLUTIONS

This report describes techniques for use in improving the design of command and control systems. All the techniques described emphasize the application of experience gained from using such systems to the improvement of later versions of those systems, either by changing existing configurations and procedures, or by adding new equipment and/or personnel. These techniques are intended to support an evolutionary design program in which command and control systems are built in increasingly ambitious stages, each stage serving as a test-bed for validating the design of its successors.

These techniques were developed in support of a program of normative exercising[1] developed by the MITRE Corporation for 473L. Under such a program, exercises of command and control systems are viewed not merely as ways for maintaining system readiness and evaluating system personnel, but as carefully controlled experiments, designed to obtain information to be used as a basis for improvements in system design and operation. Exercises are viewed in a manner similar to other equipment tests: they are intended to determine exactly what the system can do and lead to ways of improving performance. The design of such experiments or exercises faces a number of particular difficulties. Most of these derive from the special nature of command and control systems that distinguish them from more traditional military procurements. Among these are: the high cost of building and of operating (and testing) such systems; their complexity (which makes exercises difficult to write, monitor, and control); and the fact that not only the systems but also the problems with which they deal tend to be unique (so that it is difficult to generalize exercise results into descriptions of system capabilities).

All the techniques discussed in this report have the same general form. They are abstract or mathematical models of various features of command and control

---

[1] J. H. Proctor, "Normative Exercising: An Analytical and Evaluative Aid in System Design," IEEE Trans. on Engineering Management, E10 (1963).

systems. As such, they are conceptual tools intended to relate observable features of such systems. They may help one to generalize from specific exercise results, and they may allow one to predict the effects of system changes. In particularly fortunate cases, these techniques may permit one to determine the system change that will optimize some set of parameters.

APPLICATIONS

A given command and control system can be regarded as a system with the purpose not merely of accomplishing some specific job (e.g., allocating aircraft to fly medical aid in the event of an earthquake in Nicaragua) but also of making available a collection of capabilities (e.g., the ability to handle the allocation of aircraft in the event of a large variety of circumstances). When he plans and builds a system, the designer cannot really foresee all the specific problems with which the system will deal. The person testing an intermediate version of the system to determine its capabilities can test only selected problems drawn from the set of all possible problems. The basic purpose of the mathematical models described in this report is to enable system exercisers to go from various features of the results of their exercises to a description of general system capabilities, and then to use these descriptions to predict system performance for a variety of system problems.

One cannot reasonably expect a system user to be familiar with all the details of system operation or to be able to handle all the various branches of science and mathematics that are involved in system design. However, one thing that a user knows more fully than any system designer, and which he can contribute to system design, is the type of problems with which the system will actually be used. He may not know how to generalize such a description, but he can give examples. Suppose that we are responsible for the evolutionary development of some particular system. We assume that the aim of the evolutionary process is to provide the best possible product for the eventual user of the system. We conduct a particular experiment (or exercise) with the system. In this exercise, the system deals with a very specific problem (e.g., that a certain amount of medical supplies have to be flown into Nicaragua, given a certain initial allocation of aircraft to other plans, and certain time constraints). The results of this exercise tell us how well the system performed with this problem. They tell us that it took the system x long to issue the appropriate orders and that the allocation of aircraft was y short of the optimal assignment. We can substitute these results in our models and use the models to describe the existing system as a set of capabilities: the system can do a certain type of task in x time with y error.

With this description in hand we now talk to the user. Although we can explain many features of this system description to the user, it is probable that all the details and their implications cannot be communicated within the time that the user is willing or able to allocate to this job. However, we can ask the user to give us examples of the kinds of problems that he anticipates his system would have to deal with within certain time periods, or given certain types of events. We can then translate his examples to fit into the model of system capabilities derived from exercise results. Using these models, we can predict what the system would do in the case of the proffered examples. The user can then tell us where the existing system falls short of his needs.

The next stage of the process uses the mathematical models to predict the effects of changes in system performance. Given a change requested by the user (e.g., faster reaction time for a specific type of problem), there are usually several ways to accomplish such a change. Each such alternative will have associated with it both benefits and costs. Manipulations within models can allow one to estimate some of these, which can then be pointed out to the user so that he can make his decision among the available alternatives.

The use of such mathematical models to help one to "try on for size" various alternative system configurations can be compared to the use of scale drawings to test various furniture configurations in a room. One can move pieces of paper of fixed sizes and shapes around on a plan of the room to determine where one will be able to pass and what will fit into what space. The drawing and pieces of paper are not exact duplicates of one's room and furniture, but they maintain certain relationships (the relative sizes) of the originals they are modeling.

The models discussed in this report do much the same sort of thing. There are many models because each attempts only to maintain certain relationships between certain features of the system being modeled. (The model of the furniture in the room maintains size relationships and is not of much use if one wants to try out the effects of various styles of furniture or if one wants to evaluate color combinations.)

One might ask why there is no one all-purpose model. The system itself is such a model, since it has all the properties of the system. However, it is more difficult to manipulate the system experimentally than it is to manipulate a model, just as it takes more energy to move one's furniture than to move pieces of paper on a model. An actual tryout is the final test. This is the only way to find out whether the style, the color, and the locations all fit together. Trying out all one's ideas in an actual command and control system can be hard on the budget, but trying out the actual system is necessary to see if theoretical predictions are met. Models allow one to deal with such problems systematically, symbolically, and hence economically.

A mathematical model of an existing system is a collection of axioms and definitions that explicitly define the relationships between various properties of the object being modeled, together with some parameters that give the actual values of the characteristics of the existing system. The underlying axioms generally come in two parts: the first describes some well-known mathematical structure of general applicability (e.g., n-dimensional Euclidean space), and the second describes some particular object embedded into that structure (e.g., the sphere), which is the model of the particular object that is to be dealt with. Empirically derived data fill in the specific details (e.g., the diameter of the sphere is 4 in.), and the model allows us to make further assertions about particular details (e.g., the volume is a bit over 33 cu in.).

In our case, most of the axiomatic systems into which models will be embedded will be drawn from branches of discrete mathematics that are not within the traditional engineering curriculum. This may make some of the models appear more difficult than they really are.

MODELS

DEDUCTIVE INFERENCE MODEL

The deductive inference model, described in Appendix I, deals with a command and control system as a system whose fundamental role is to make deductive inferences. The notion of a deductive inference is generalized to include any process that can be described as applying explicit rules to strings of symbols. Such strings of symbols include not only statements of facts but also statements of values and features of the command structure. The processing of such strings is used to model activities such as: the processing of message inputs for incorporation into the memories associated with the system; the determination of the logical consequences of events described in inputs, as they affect the various elements under the control of the system; the selection of some particular course of action from among those possibilities allowed within the given command structure and authorized under the system's mission; and the translation of these deductions into appropriately routed outputs of suitable format.

This model was originally intended as a tool for use in the design and control of normative exercises. In that application it is desirable to allow only one possible solution and to give a system increasingly strong hints as to the nature of the proper solution, to guide it back on the normative path when it deviates from it. This model can also be used as a tool for describing a command and control system as a system which performs such deductive inferences. Thus, it is a general tool for the application of exercise results to system improvement.

In Appendix I, the basic view of system operation which underlies this model is described and an effort is made to justify it. Then the basic elements of which the model is constructed are presented. Finally, a simplified application is described.

INDUCTIVE INFERENCE MODEL

The inductive inference model of Appendix II looks at a command and control system as a system that does something beyond applying given rules to given symbols. In making a deductive inference from a set of strings $S_1, \ldots, S_i$, one applies some given set of rules of inference ϕ to those given strings; the result of applying these rules gives a conclusion C. In inductive inference, the rules one is given are not adequate for the derivation of C. Rather than rules, one is given a series of examples. (The examples are usually given in the training period and constitute "experience.") From these examples, the system (or the people in it) have to determine the nature of the rule that is to be applied. Only then can they apply it in the manner described by the model of Appendix I. The inductive inference model provides a basis for making explicit the way in which certain factual aspects of past experience are brought to bear on current problems in a command and control system.

We assume a system whose mode of operation is completely explicit, with the task of predicting symbols on a tape. Such a system must make assumptions about its environment beyond those that it can derive empirically from such a tape. The relationship between various general inductive strategies and such assumptions is investigated. We also investigate the relationships between these features and the amount of training required to achieve a given level of performance.

Although some interesting results are derived, such a model is still somewhat removed from practical application to command and control systems. Nevertheless, this model appears to have some potential as a medium for the eventual automation of some aspects of such systems, and for solving some problems in adaptive pattern recognition.

VALUES AND DECISION MAKING

Appendix III is concerned with the relationship that exists between human values and decision making in a command and control system. We were encouraged to undertake this study because we believe that a determinable relationship exists between value and decision, and that the determination of this relationship can lead to some useful training concepts for command and control system decision makers. We foresee the possibility of predicting a person's decision behavior on the basis of our knowledge of his value spectrum. This notion leads us to speculate about the possibility of being able to select good decision makers by measuring their value spectrums and of training decision makers by training candidates in the desired values.

In this phase of the study we confined ourselves to a search for valid and reliable measures of value and representative tools for determining a person's decision making pattern. Value and decision questionnaires were constructed and administered to an experimental group. The resultant data were treated statistically and analyzed to determine the extent of the predicted relationships.

SEMANTIC MODEL

The semantic model (Appendix IV) deals with the representation of an image of the external world within the system. An analysis is made of existing theories for representing such an image, and these theories are found to be inadequate. Some abstract features of the ways in which human beings handle meanings are derived, and an explicit structure that has these properties is briefly described. This model is also somewhat removed from practical application in the exercise and evaluation of command and control systems, but may eventually provide a basis for measuring some of those aspects of command and control system behavior which involve meanings.

FINITE AUTOMATON MODEL

The finite automaton model of Appendix V is based on the theory of finite automata. It describes the relationships between the systems being controlled during an exercise and the system attempting to control the exercise. It relates the amount of information obtained by the controlling system, and the amount of information this system gives to the systems being controlled, to the effectiveness of control. It also relates various parameters of the exercise to the efficacy of the exercise as a learning experience for the command and control system. It may be used to study the effectiveness of different command and control system strategies.

This model is perhaps best used for the study of relationships between values of parameters and the resultant behavior of the command and control system, both during an exercise and during actual operations. This model may be useful in anticipating problems in control that might arise during an exercise, given the exercise design. It may help in planning exercises that will be more satisfactory learning experiences, and may be useful in the study of overall command and control system strategies. This model is particularly well suited to computer simulation.

SUGGESTIONS FOR FURTHER INVESTIGATION

The five models described in this report are tools that may provide the basis for a precise and scientifically sound methodology for use in many phases of command and control system exercise and evaluation. They may also make possible the automation (or at least simulation) of various phases of command and control system activities.

Each of these models appears to merit further investigation. The value of conducting such investigations should probably be considered separately for each model. Since these models appear to have applications that go beyond the exercise and evaluation of command and control systems, they should probably be investigated independently of such an application.

The model of Appendix I appears to have some value as a basis for the automatic generation of exercise problems and the automation of command and control system problem-solving activities. The model of Appendix II might have applications in pattern recognition and the automation of inductive inference. The model of Appendix III might have applications in the training of command and control system personnel, and in psychology. The model of Appendix IV may have applications in computational linguistics, and the model of Appendix V appears to provide a basis for simulating both command and control systems and human psychological processes. Many of these applications probably deserve to be investigated.

APPENDIX I

DEDUCTIVE INFERENCE MODEL

INTRODUCTION

In this appendix we consider a command and control system as a system for providing the reasoning power that lies behind optimal utilization of the resources of a military command for some established purpose. This purpose might be the utilization of interceptors for the defense of cities or the allocation of fuel oil to depots to maintain readiness for the execution of plans. In constructing a model (or class of models) of this aspect of a command and control system's activity, we shall use techniques from mathematical logic. In using an established mathematical technique on a particular problem, one attempts first to discover an appropriate set of independent and dependent variables, and then to describe a model for relating the former to the latter. In our case, the inputs to the system will constitute the independent variables; the outputs will be the dependent variables to be predicted* by the model.

The variables related in this type of model cannot easily be related to numerical values. They cannot be measured in the traditional sense. The precision of the model lies not in making everything measurable and numerical but in making everything explicit and definable in terms of observable features alone.

We are trying to do something more general than defining a mathematical model for a particular system. We are trying to define and justify a certain type of mathematical apparatus as adequate for describing this type of system. This is not an ad hoc procedure. We chose our type of apparatus (mathematical logic) because it had been well developed. We took it as far as we could, but in applying it, it became clear that certain things had to be left out because the apparatus for dealing with them was inadequate. These things were then subjected to separate investigation, which led to the investigations described in Appendixes II, III, and IV. Thus we have not tried to squeeze command and control systems into a certain type of mold, but rather to see how far we could go with a particular method.

---

*Predictions will be definitions of the set of all things that a given system could logically do under given circumstances.

In this appendix we first show one specific way of looking at what we consider important in the behavior of a command and control system. We do this by stating a series of increasingly restrictive assumptions, beginning with a general characterization of a command and control system, stated and discussed in natural English. We end with a more rigorous set of assumptions, which could be stated as formal axioms. We are concerned not with the mathematical investigation of these axioms but with a validation of their adequacy as descriptive of a command and control system's activity, particularly as this bears on normative exercising.[1] The manner in which this adequacy was determined is discussed briefly at the end of the appendix.

BASIC ASSUMPTIONS OF THE MODEL

GENERAL CHARACTERIZATION OF COMMAND AND CONTROL SYSTEM ACTIVITIES

A command and control system is an information-processing system within a military command structure. It receives orders from, and issues information to, higher commands and, in turn, issues orders to, and receives information from, lower commands. (See Figure I-1.) Between these input and output operations

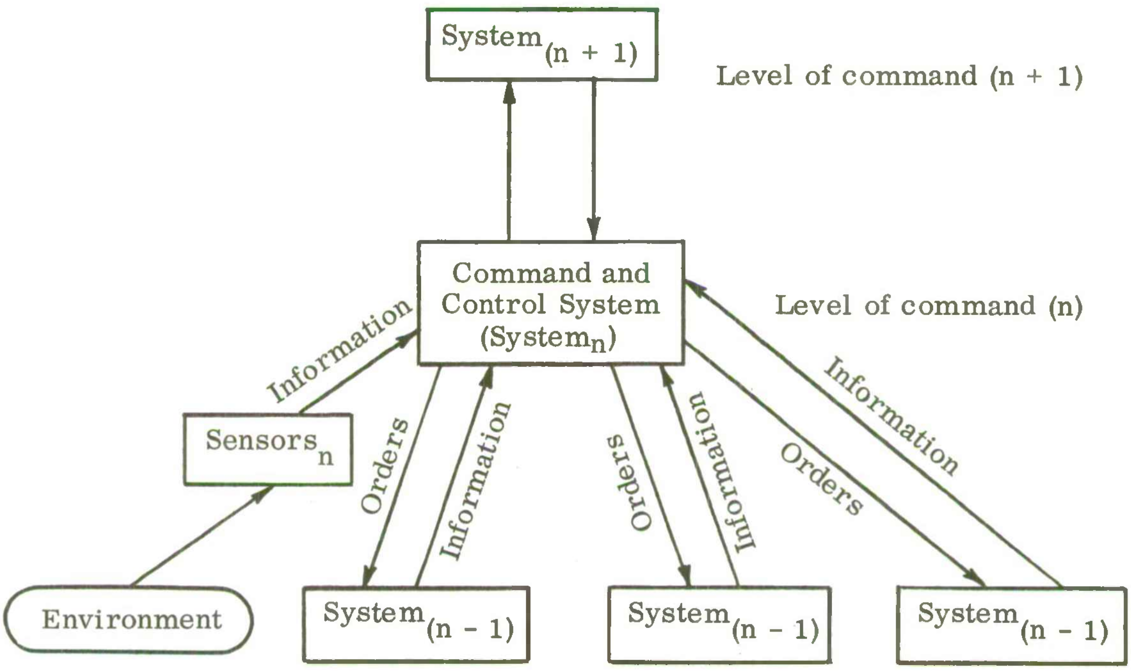


Figure I-1. A Command and Control System Within a Military Command Structure there are various types and amounts of activities within the system. These activities vary from system to system and from situation to situation. In this appendix we shall treat one aspect that most systems appear to have in common: they are information-processing activities.

We will be concerned with the deductive aspect of this information processing. Given a situation, given a mission or orders from a superior command, and given the military command structure that determines the kinds of orders a system can issue, there is a "best" course of action for such a system. If the system had unlimited time, it could lay out all alternative actions that it might take and evaluate them according to the criteria set by its mission or superiors. It could then select from these the one that was best.

In actual practice, a command and control system does not work in this manner. Although the set of alternatives available to it at any time may be determined by the facts given, the system usually does not have the time to examine and evaluate all of them to select an optimum. Ingenuity or clever mathematical techniques are required to select the best course of action within limitations of time and space.

We are not concerned here with the techniques that bring the finding of optima within the available time constraints. We are concerned only with describing the class of all possible actions that a system might take, given a certain situation.

There are a number of reasons for being interested in providing such a description. Our original motivation was based on the requirement to support a program of normative exercising.[1] To implement such a program, it was necessary to know what a given system might do with a sequence of inputs in order to provide for possible system responses in the preparation of an exercise. However, the ability to describe all those things that a system might do, given a situation, has a number of other applications. For the purposes of this report, the most important application is a description of the capabilities of both existing and planned systems that can serve as a vehicle for using exercise results in system design. Describing all the things that a system might do is a preliminary step in the investigation of algorithms for seeking optima within realistic time constraints.

The information received by a command and control system may change the image that the system has of the status of subordinate systems and their environment.

A command and control system issues orders in order to produce some change in the lower commands. This change is sought in order to produce a situation that meets certain criteria that have been established by higher commands. For example, a system may find a certain track on its radar that changes its view of the nature of the environment of the aircraft under its control and of the cities it is to protect. As a result of this change in image, the system may issue an order to launch interceptors. The purpose of this order is not to create some state in the aircraft headed for the interception, but to protect some potential enemy target from attack.

RELATIONSHIP TO THE RESOURCE ASSIGNMENT MODEL[2]

The ability to relate both the information it receives and the orders it can issue to the effects that will ensure is the basis of the reasoning power of the command and control system. This ability depends on:

1. The kinds of information that the system receives (or can obtain) about the systems being controlled and their environment.
2. The ability of the system to process this information.
3. The kinds of orders that the system can issue.

It is customary to restrict the role of the command and control system designer to the development of new machinery for (1) and for the development of procedures and machinery (or programs) for improving (2). However, it is clear that the reporting procedures that influence (1) and the military command structure that determines (3) are also important variables.

In our interim report[2] we described a formalism for dealing with (2). The purposes of this appendix are (a) to describe an improved version of that formalism and (b) to extend this formalism to include various aspects of (1) and (3). We have been led to the development of this extended formalism as the result of an effort to apply our previous formalism to the analysis of exercises of the 473L system provided for us by the MITRE Corporation.

One of the important problems facing a system like 473L is not the solution of problems, but rather their timely recognition and identification. This aspect of system performance requires a formalism that includes provisions for handling (1) and (3). A second result of our experimental application of the resource assignment model is an increase in the simplicity and generality of the model.

FROM INFORMAL ASSUMPTIONS TO A FORMAL SYSTEM

Our model is based on a number of assumptions that lead to an increasing specification of that model.

> Assumption 1. A command and control system is an information-processing system.

This assumption has two roles. First, it defines the domain of investigation. That is, in our formal development we shall not consider anything that a command and control system does that cannot be described as information processing. This is much like saying that, from a point outside a line, one and only one line can be drawn parallel to the given line.* We do not say that this is true for the world, but only that it is true in the space we are going to study, or that we are going to assume it true and investigate the consequences of this assumption.

The second role Assumption 1 plays is to assert something we feel to be true. We want to select a postulate that usefully characterizes a system. (To say that a command and control system is an information-processing system is trivial in the sense that information processing is an observable feature of such systems. What we are saying is that such systems can be profitably regarded as systems that do nothing else, even though many things that go on in them cannot easily be described as information processing. For example, fuses blow and people tire. The assumption tells us to ignore these things. In making this assumption, we claim that a useful model will ensue .)

> Assumption 2. Information processing can be described as string† manipulation according to rules.

This assumption restricts the tools we are going to use in dealing with information-processing aspects of the command and control systems. It says that

---

*Euclid's fifth postulate.

† For the purposes of this appendix, a string is a sequence of symbols, often including the blank. Thus the sentences of this appendix are strings.

information processing can be described as the processing of strings of symbols, that the inputs and outputs of information-processing systems are strings, and that the behavior of information-processing systems is rule-governed.

A string is a sequence of letters (or symbols) from a finite alphabet. If the alphabet is (0, 1, 2, 3, 4, 5, 6, 7, 8, 9), then the following are strings on that alphabet: 90, 10948, 9. If the alphabet is (A, B, ... , Z) then the following are strings on that alphabet: AAA, AIRCRAFT, ALTITUDE. Manipulation of strings is usually defined in terms of a set of functions that map strings or sets of strings into other strings or sets of strings. Usually we require that these functions be recursive, but this is not necessary. An example of such string manipulating functions is the one that orders a set of strings alphabetically. Its input is a set of strings, and its output is a set of strings ordered in a certain way. Another function of this sort can take pairs of numerals (e.g., 2 and 51) and produce a single string (53) that denotes their arithmetic sum.

Both of these functions are defined in terms of operations that work on the strings letter by letter: the first working from left to right, the second from right to left. The result of each operation is defined in terms of (a) the letters being considered and (b) the result of the preceding operations. Although such definitions deal with a potentially infinite set of strings, the definitions themselves are finite. Also, although these definitions can handle strings of arbitrary length, the operations that define them operate on segments of restricted length. These operations are defined on explicitly (mechanically) recognizable features of the strings, not on indefinite ones such as the meaning of the string.

> Assumption 3. A command and control system exists within a military command structure that defines (a) what such a system can and cannot do and (b) its mission.
>
> Assumption 4. A command and control system has an image of (a) the systems being controlled, (b) their environment, and (c) certain laws that extrapolate these into the future. This image determines what such a system actually can do or thinks it can do.
>
> Assumption 5. Assumptions (3) and (4) combine to produce problems. The main purpose of a command and control system is to identify and solve such problems.

A problem is defined by a system's position in a command and reporting system (Assumption 3) and by its image of the world (Assumption 4). A problem exists for a system when (a) some feature of the representation of the world situation is incongruent with the system's mission and (b) it is possible for the system to improve that situation by issuing an order within the command structure.

Whether a problem is so identified or solved depends on two things: (a) the system's image of itself (or those features of itself covered in Assumption 3) and (b) the system's image of the external world. Both of these can be described as sets of strings (by Assumption 2) and as the results of processing other strings. Much of the system's image of itself exists within the minds of its personnel, which is hard to make explicit since the theory of the workings of the human mind are not sufficiently developed. Some of the image lies in the more explicit rules of procedure, which might possibly be described in terms of string manipulation. But in the formalism to be described here, we will ignore this also. That is, we shall assume perfect understanding of the system's mission by the system.

Errors may occur also in the production of the system's image of the world. These errors, which may be of concern to the designers of the system, are of two sorts: (a) omission and (b) erroneous information.

As shown in Figure I-2, the construction of the system's image of the world occurs in two phases: (1) the acquisition of information about the environment and

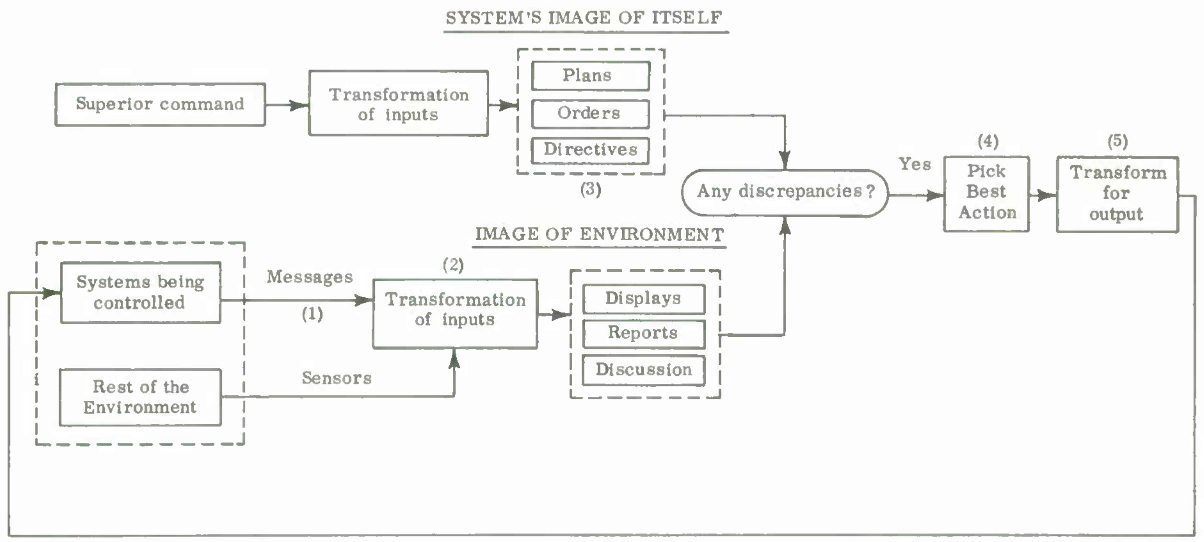


Figure I-2. A System's Image of Itself and of the World

(2) the manipulation of this information within the system. Three other steps follow these: (3) the system must, in some manner, bring its image of itself to bear on the image in the world in order to discover discrepancies that call for system action; (4) it must select from the (possibly infinite) set of possible system actions that action it considers to be the best possible,* and (5) it must format the results of that processing and output them in appropriate representations.

BASIC ELEMENTS OF THE MODEL

Our model will be built up out of certain abstract elements. These elements will include (a) sets of strings, (b) orderings on sets of strings, and (c) functions from and to strings and sets of strings.

SETS OF STRINGS

Sets of strings are generally infinite. Since mathematics and science require finite specifications of the objects with which they deal, these objects are usually defined in terms of finitely statable rules that can generate infinite sets. One first introduces some finite set of symbols, which is referred to as an alphabet. (Examples: (0, 1, 2, 3, 4, 5, 6, 7, 8, 9, +, -, =), (A, B, . . . , Y, Z, , .))† One then proceeds to define a sequence of rules to generate strings. The manner in which this is usually done evolved to serve the needs of mathematical logic. Because our application is somewhat different from the usual, we will change it slightly.

The first step is to define the set of all possible strings. The members of this set are all the results of putting together or concatentating members of the alphabet.‡

*The best possible action may be that of doing nothing.

†In principle, finite sets can be defined by listing their elements. In denoting such a set we surround a list with parentheses and separate the symbols by commas. The first of these sets includes the symbols 5 and +, but not the comma. The second set could be listed exhaustively too, but since this would take too much space we mark the ellipsis in the standard way. R and S are members of the second alphabet. So is the blank, the next-to-last item in the list.

‡We will distinguish the two ways in which one can refer to the members of this alphabet as "tokens" and "types." A token is an occurrence of a given letter; a type is the class into which all those occurrences fall. In the word AIRCRAFT there are two tokens of the letters A and R. Both occurrences of A are of the same type.

The first alphabet permits such strings as 4130 + 4 ---, 54, and + - + - +; the second permits strings such as ASECIRS, THE AIRCRAFT, and
The last is a string of blanks.

Next, one provides a set of rules that pick out of this underlying set a set that is supposed to include those strings capable of having meaning. In the case of the first alphabet, one might specify a set of rules that allows such strings as 90 - 34 = 6, and 23 + 3456 - 3 = 67 - 4. In most treatments a final set of rules is now added to define those statements supposed to be true. This rule might allow us to generate such strings as 2 + 3 = 5 and 2 - 2 = 0, but not 2 + 2 = 5. Figure I-3 shows these three steps in a standard description of a formal system.

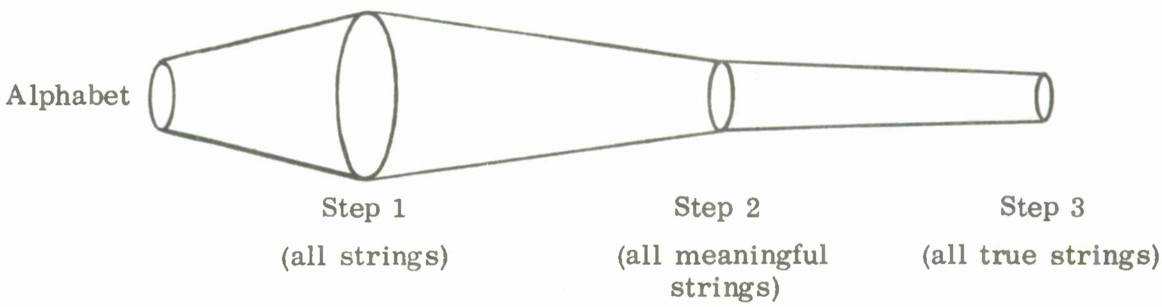


Figure I-3. Steps in Describing a Formal System

Since our treatment is going to be more general, we will often need more rules to permit us to select more sets of strings at the stages of the process shown in Figure I-3. We will need to define sets that denote all the allowable orders a system can issue, and to define rules for selecting from these orders those strings that denote the commands it must issue. We will need such rules in addition to the rules that select sets of strings denoting facts. Furthermore, our definitions will have to be more dynamic because our sets of strings denoting truths will be functions of continually changing sets strings, which will denote the inputs to the system.

In our treatment of strings here, we will be general and abstract. However, we will use illustrative examples. We will not distinguish different alphabets from which strings can be drawn, although we will assume all of them to be finite. For example, we will ignore the difference between the word "speed" written as the five occurrences of letters from the second of our sample alphabets (SPEED) and that same word written as the five pairs of letters 22, 47, 65, 65, 64 from our first alphabet (which is a transliteration of how that same word would appear on BCD tape).

Sets Defined by Length

An alphabet is a finite set of distinct symbols, or types of symbols in the sense of the last footnote on p. 16. A string on an alphabet A is any result of concatenating tokens of symbols in A. (The null string, which consists of no letters from A, is usually excluded.) The length of a string is the number of occurrences of symbols that appear in it.

Three types of sets of strings can be distinguished by length:

(L-1) A set of strings may contain only those strings on the given alphabet whose length is less than some given n.

Examples:

(a) The set of all possible words in a computer with fixed word length.

(b) The set of all integers less than some given integer.

(c) The set of numbers between 0 and 1 with accuracy limited to some fixed number of decimal places.

(L-2) A set of strings may contain only strings of finite length.

Note: Sets of type L-1 must be finite since there is, by definition, only a finite number of letters in the alphabet; strings of type L-1 are limited in length. However, sets of type L-2 can be infinite because, although each of their members is finite in length, there is no upper bound on this length: given any string, there is always a longer one. Consider the strings generated by the one-letter alphabet (1): 1, 11, 111, 1111, ... Although each of these strings has a finite length, the set is clearly infinite.

Examples:

(a) The set of all possible books that can be written using the English alphabet (including punctuation marks).

We include all books that might be produced, whether they make sense or not. If we restrict this set to books that make sense, we are dealing with a set that is properly included in this one, but which requires more powerful machinery for its definition than we have discussed so far.

(b) The numerals used to denote the non-negative integers, the generating alphabet in the usual representation being the alphabet (0, 1, 2, 3, 4, 5, 6, 8, 9.). The rational numbers can be represented by pairs of strings from such a set, using the same alphabet as in the preceding example.

(c) The set of all possible messages, including those that are garbled, encoded, and otherwise mangled.

(L-3) A set of strings may include strings of any length.

Note: Such a set includes infinitely long strings, which are ignored in many treatments because they have little practical application in the usual concerns of mathematical logic. (Since they can denote the continuum, they are often introduced into logic courses in order to prove that the continuum cannot be enumerated.) There are more strings in sets of type L-3 than there are of type L-2 since any set of strings of type L-2 can be enumerated.

Examples:

(a) The set of non-terminating decimals used to represent the real numbers between 0 and 1 (if one leaves out the decimal point).

(b) The same set used to represent sensor inputs.

(c) The set of arbitrary strings on the alphabet (0, 1, 2, 3, 4, 5, 6, 7, 8, 9) used to represent the real numbers.*

---

*We have been assuming a one-way infinity only, and we have been using the beginning of the strings as a marker to denote the decimal point. If we want to denote a continuum without bounds, we might include a single mark (.) which we would allow only once in a string. The additional rule that one of the symbols in the alphabet can appear only once in the string goes beyond the limits we have discussed so far. However, it is easy to use strings of type L-3 to denote the points of the real line by having the digits to the left of the decimal point appear on the even positions and the ones on the right appear on the odd positions. Thus, 245.987 would be written 9584720000 ... .

One reason for being concerned with sets of type L-3, although they cannot be realized, is that they provide a model of sensor inputs when these can take arbitrary values in an interval or on a display surface. The reading of such values by an operator requires that they be truncated in some manner, constituting a mapping of a set of members from a set of type L-3 to members of a set of type L-1. The storing of messages in the computer of a command and control system can be regarded as a mapping from strings drawn from a set of type L-2 to strings drawn from a set of type L-1.

Sets Defined by Rules

Given strings of the type we have been discussing, we will usually want to thin them out. In the usual presentation (Figure I-3), thinning-out procedures are defined in two steps.* In this appendix, however, we will amalgamate these two steps. Given a set of strings of one of the L types, we can "thin" them out using the following types of rules:†

(R-1) We permit all possible strings within the limitations of length.

Note: We include this "null rule" because the string sets selected by length define the universe of all possible strings, while strings in this section are supposed to thin out these sets to include only the strings that one can ordinarily expect to encounter. (In a natural language, such a procedure might thin out the set of all combinations to the set of all the meaningful, or at least pronounceable, strings.) There is no reason to exclude the set of all strings from being identical to the set of all expected strings.

Examples:

(a) The set of all points on a square display surface that are distinguishable to within some limited accuracy.

---

*The first thinning out selects the set of well-formed formulas, which must be recursive. (One must be able to recognize a well-formed formula when one sees it.) In most formal systems, however, the set of theorems is not recursive. At best it is recursively enumerable. (The fact that this is true for the first order predicate calculus is known as "Church's theorem.") The second thinning out selects theorems, but it cannot be recursive in such cases.

†These are the procedures used in steps 2 and 3 of Figure I-3.

(b) The possible contents of the words of a given computer (without error-correcting bits).

(R-2) We permit only those strings given by a finite list.

Examples:

(a) The set of all words that name categories in a data base.

(b) The set of names that can be looked up in a telephone book.

Note: Such strings often appear in tables that map them onto the words of another set. The sets of R-2 are distinguished from the following:

(R-3) We limit the members of the set to strings that can be recognized by applying some given mechanical procedure.

Note: For strings of type L-1, subsets drawn out by rules of the types R-2 and R-3 are, in principle, identical. However, there usually are intuitive distinctions between them. If one selects from the names of all Boston residents (1) all names containing less than 20 letters, and then (2) the class of all pronounceable names beginning with the letter A and containing no more than five letters, one is more likely to describe the first by presenting a list (method R-2) and the second by presenting a rule (method R-3).

Examples:

The sets of equations that can be generated from the letters of the alphabet given on p. 16, or the set of well-formed formulas in a mathematical system. It can be argued that the sentences of natural languages can be described in this way; otherwise we would not be able to recognize membership in them. Given such a set, it is not always easy to find a rule that actually recognizes members of it. Some mathematical problems that are still unsolved consist of trying to devise such procedures.

(R-4) We limit the set to strings that can be generated from the initial set by some rule.

Note: It may be possible to generate a set of strings mechanically without being able to recognize mechanically whether a given string is in that set. There are many mathematical systems whose theorems fit into this category without fitting into the category of R-3 (see first footnote on p. 20). We need such sets to describe the notion of "logical consequence."

Example:

The theorems of the first order predicate calculus.

We will not need to add machinery to define the set of all strings true in some particular interpretation. The set of exactly those strings that represent the theorems of elementary number theory cannot be recursively enumerated and therefore cannot be described by the four rules presented above. (This is the consequence of Goedel's incompleteness theorem.) Such sets do not seem to be required for the purposes of this appendix. However, number theory is indicated for the purposes of Appendix II.

SETS OF SETS OF STRINGS

For our model we will need various sets whose members are themselves sets of strings defined in the above ways. The use of such sets will enable us to impose orderings on sets of strings to represent such things as command structures and relations between parts and wholes. Sets of sets of strings will also enable us to define functions on strings and other types of mappings that will provide a sufficient basic structure for describing the deductive information processing of command and control systems.

Relations

Sentences such as " C124's can carry more cargo than C130's." and "The distance between Boston and New York is less than the distance between Boston and Washington." assert relations. The first can be interpreted as representing a relation between the strings C130 and C124, while the second represents a relation between the strings Boston, New York, and Washington (which in turn denote the cities Boston, New York, and Washington).*

---

*The former are strings, and the latter are cities. In this report we are not generally making a notational distinction between a string used as its own name (Boston, used as the name of something that begins with a B) and the same string used as the name of an external object (Boston, used as the name of a city having a million inhabitants).[3] A reason why we are trying to restrict ourselves to string handling alone in our model is suggested by the difficulty one would have in getting the second of these Bostons into a command and control center, and the ease with which the former can fit.

Relations can be treated as functions that map ordered sets of strings into the values true and false. Such a function might represent the relation "greater than." In that case it would map* [5, 3], [9, 5], [34, 8] into the value true, and the pairs [4, 78] and [3, 5] into false.

Orderings

In dealing with relations, we are concerned not with specific relations but rather with sets of relations that share certain abstract properties. Since the classifications that follow are standard, we emphasize the applications rather than the characterizations.

> (O-1) A relation R partly orders a set if R is reflexive, anti-symmetric, and transitive on that set.†

Relations that have this property are classically the relations "greater than or equal to," "included in" (as used in set theory), and "included in" (in the physical sense). In our case we will also use relations of type O-1 to represent command structures.

A relation that imposes such an ordering can be defined in a number of different ways:

1. It can be given explicitly as a table of organization or a tree.

2. It can be given by a rule. We would state a rule to represent the relation of "being located in" a physical space (usually represented on a map). The facts that Boston is in Massachusetts and Massachusetts is in the United States would be represented by referring both to physical locations (in terms of latitude and longitude), and by defining the relation as the result of a computation on these coordinates.

3. It can be given by a combination of the methods of (1) and (2). We would want such a representation if we were to deal with the locations of individual

---

*Parentheses will be used to denote sets. Brackets will be used to denote ordered sets.

†A relation R is reflexive if R(x, y) is always true for x = y; anti-symmetric if R(x, y) always implies that R(y, x) is false unless x = y; and transitive if R(x, y) and R(y, z) always imply that R(x, z).

personnel. We might store the location of a particular squadron in manner (1) by associating its name with the string Hanscom Field, which in turn would be associated with another string denoting its latitude and longitude. The facts that a certain aircraft is assigned to that squadron and that a certain airman is assigned to that aircraft could be stored in manner (1). The fact that Hanscom Field is in Massachusetts can be derived by a rule (2). The transitivity of the relations now allows us to use this single representation and to derive the fact that the aircraft is based in Massachusetts, as well as other facts.

The reasons for choosing one sort of representation rather than another are usually based on a requirement of simplicity. Things that tend to change over time and/or that are not easily described by rule are stored in manner (2). Relations that connect points in two types of spaces are generally stored as simple functions. These are often in the form of dictionaries or finite sets of ordered pairs. For example, these might map the string Hanscom Field into a string that denotes its coordinates.

(O-2) When a relation holds between every pair of members of a set (as well as having the properties of O-1), we say that it simply orders the set.

Note that trees (other than degenerate ones) do not have this property. Given a tree of the form

$$\begin{array}{ccc} & A & \\ / & | & \backslash \\ B & C & D \end{array}$$

the relationship it diagrams holds between A and the other members, but not between B, C, and D. It is easy to see that a tree with this property must be degenerate in the sense that it becomes a linear ordering (i.e., it has no branches). Such ordering will appear in those sets that denote numbers.

Relations of type O-2 can be defined either by lists or by rules. If the domains of such relations are infinite, they must be given by rules since we make no allowances for infinitely large tables.

Simple orderings are usually imposed by assigning numerical values to objects. In our formalism we assign functions that map alphameric strings (e.g., the cargo capacity of a C130) to strings that have dominant numerical parts (35,000 pounds). Usually the strings into which the mapping occurs are numerical with associated units (e.g., pounds). These units have not only a scale (pounds) but also an associated dimension in the usual M, L, T sense. The dimensions often allow units to be translated into other units, but even the functions that translate units within the same dimensions into each other can be quite complex. This complexity occurs when one asks whether a certain object can be flown on a particular aircraft. If one is clearly within weight limitations, one may still have to worry about rather complex relations between numerical values of the dimension (0, 1, 0): e.g., will it fit through the door?

(O-3) A relation R imposes a complete partial ordering on a set if R meets the following condition:*

$$\bigcup_{x} - (\exists z)\ (R(x, z) \cdot R(z, y)) \;=\; y$$

Completeness in this sense will be an important property when we are using such orderings on various aspects of the forces under the control of a given command and control system. Completeness in this sense will correspond to completeness of information within the system's representation. For example, if a complete ordering of the command structure includes a given command, then that ordering includes all the immediately subordinate command units (if it lists any at that level).

We shall be concerned with one relation that is irreflexive, anti-symmetric, and intransitive. This is the relationship of class membership, denoted by ϵ.

We are concerned with separating these various types of ordering in our model. Many of the things that we would like to say about a large variety of specific relations of a given type need be said only once in a given representation if the relations are sorted by type. By singling out the relation "is located in" as a relation of type O-1, we can derive the fact that if A is located in B and B is located in C, then A is located in C. This fact is obvious to a human being and need not be sorted separately.

---

*The case where the relationship R is that of whole to part is referred to as a complete resource net.[4]

Functions

Functions defined over sets can be looked at as sets of ordered pairs that take their first members from the domain and their second from the range. These functions allow us to relate one set of strings to another. One such function relates the string Hanscom Field to its geographical location, which is a pair of numbers. We also need a function that relates the locations of two airfields and the characteristics of particular aircraft to the flying time from one of these airfields to the other. Generally, the more complex such functions are in terms of computation, the more accurate they are. One of the things that a system designer has to determine is how accurate a system he needs; increases in accuracy usually require an increase in computation time.

Functions on sets can be given exhaustively by list, or they can be given by an algorithm that computes them, given the values of the arguments. The reason for separating out the functions is that, because the same function is used over and over again, separating it yields a simplification of the representation.

Such a separating of the various functions can increase the power of a system, but it also increases the difficulty of constructing it. Usually these functions are only partially separated. This is done by building the system in two parts. The first part is a programming system, which contains the basic functions, and the second consists of the actual programs written in terms of those functions. By separating out the basic elements, we have constructed (or listed the specifications for) a programming system to underlie the particular languages that will be used in describing command and control systems. We have specified the basic categories to be used in constructing such descriptions. (In the sections Sets Defined by Length and Sets Defined by Rules, we listed basic data elements. In the sections Relations, Orderings, and Functions, we listed the basic materials for macros, or basic system functions.)

AN EXAMPLE

INTRODUCTION

When applying the abstract structures described in the preceding part of this appendix, one must select sets of strings that represent particular things, particular functions with which to describe manipulations, and so forth. How this selection is made depends on the characteristics of the system being modeled.

In order to suggest how such selection might be done, we shall sketch such a selection for a restricted example. There are at least two reasons for limiting ourselves in this way: (1) by using a restricted example, we are able to highlight essential features, and (2) by choosing an example that is admittedly partial, we lessen the likelihood that the reader will think that this is the only way in which this formalism can be applied to the description of a command and control system.

Although the example is limited, the basic notions that underlie it have been derived from an extensive effort to apply the basic structures of the preceding section to the description of two exercises of the 473L system. The results of this effort have suggested that the collection of structures described in the preceding section is adequate for describing an exercise when one wants the model to define the set of all possible solutions, given a description of a problem. However, although we found this machinery basically adequate, we also found it unwieldy. It is almost impossible to make the necessary computations by hand. In our model we represented most of the processing that goes on in the modeled system, including the processing that goes on in the minds of system personnel. Even after one sifts out the parts of an exercise that are not relevant to the intended problem, the processing is complex. This complexity led us to reduce the number and variety of elements required.

PROCESS TO BE DESCRIBED

The information processing that occurs in problem identification produces a sequence of strings that state a particular problem and state that it is a problem for the system in question. The information processing in problem solution results in a sequence of strings that describe what is to be done.

The problem with which we will deal consists of a world situation that will lead our sample system XXXL to anticipate that a given plan P is probably going to be executed and that, given the current allocation of resources, the execution of P will be hindered by a shortage of C130's in command YYY. The solution of this problem will consist of selecting a particular group of C130's from command ZZZ and reassigning them to YYY in time for their use in plan P.

This problem is quite simple to solve. Our purpose in describing its solution is to demonstrate the kind of machinery we will use.

ELEMENTS OF THE PROCESS

Mission

We might describe the mission of the XXXL system as that of handling resource shortages when this requires obtaining resources from some command other than the one responsible for the plan. In translating this mission into a formal statement, we must make explicit the elements of the mission statements that bear on other aspects of problem solution. The XXXL system will have some sort of a plan file F, and it can be assumed to be responsible only for the plans that are actually in that file. Since the file is dynamic (plans can be removed from it and added to it), we will refer to the file as F_t when we want to make the time parameter explicit.

The contents of F_t are functions of inputs prior to t. In general we do not expect a plan that is input at t′ to be in $F_{t'}$ since the process of entering a plan into the file takes time. (In systems that have computerized file plans, the entry of some given plan into F will require formatting, card punching, and other transformations.)

Plans

A plan is a set of strings that meets certain syntactic requirements. A change in a plan is an operation on that set of strings. An execution of a plan can be regarded as a function whose arguments are the plan and the system's representation of the status of the world, and whose values are sequences of such representations indicating the changes that occur as the result of plan implementation. When the value of this function does not meet certain criteria given either in the statement of the system's mission or in an order from a superior command (e.g., specific time constraints), a problem exists for the system. In order to make explicit the elements of a plan needed to define the function discussed above, one must specify the syntax of a language for writing the descriptions. Such a definition can be presented in Backus normal form.[5] Let us assume that we have available for the construction of plans all the usual alphanumeric symbols. Since we want the name of the plan to be specifically identifiable, we place it at the head of the string that represents the given plan. In order to select a particular set of strings (e.g., P followed by an arbitrary numeral) as allowable, we might write the following sequence of syntactical rules in Backus normal form:

<Digit> :: = <0, 1, 2, 3, 4, 5, 6, 7, 8, 9>

<Number> :: = <Digit> | <Digit><Number>

<Plan Name> :: = P<Number>

A function defined by a plan number can now be defined in terms of an explicitly defined segment of the string that represents the plan. We will henceforth assume the existence of complete syntactic descriptions (in the above form) of the various languages with which we are dealing; we will not present them further except to indicate some of the methods of simplification we have discovered.

One segment of the string associated with a given plan denotes its resource requirements. Assume that this segment is set off by the marker REQUIREMENTS at each end. Requirements can be of various types. They may designate a particular unit, or a particular quantity of a particular type object (e.g., 30 C130's). We will write strings of the second sort with a solidus to separate the string denoting quantity from the string denoting type of unit. For example, to indicate the need for 30 C130's in a particular plan P, we place the string 30/C130 in the part of the plan that denotes resource requirements. The requirement for 30/C130 may imply other requirements (e.g., fuel). These other requirements may be the result of applying a separate rule, which need not be repeated for every plan in which a requirement for C130's appears.

We also need to represent given quantities of materials. We can do this by permitting a statement of units and dimensions* parenthetically after the quantity and before the solidus. Thus 3 tons of sugar can be denoted by the string 3 (tons)(1, 0, 0)/sugar.

---

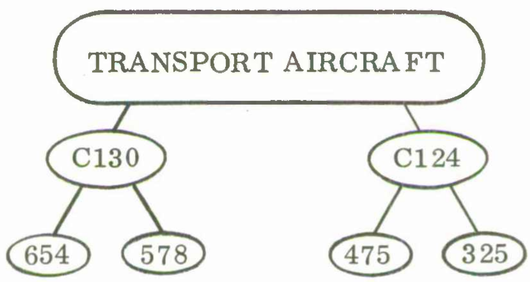
*Mass, length, and time.

A plan may also require resources that are specified only by a requirement that they meet certain conditions. And a plan may allow for alternatives. For example, P may call for 30 C130's, or equivalent troop-carrying capabilities if C130's are not available. To handle this situation, we need to extend our notation in several directions. First, we need machinery for handling a sequence of alternatives, with certain alternatives preferred. Temporarily, we shall use ϕ to indicate the string that denotes the allowable substitutes for the 30 C130's. We can use conditional expressions of the form:

$$((30/C130),\ (\phi)) \qquad \text{(S-1)}$$

The requirement expressed by S-1 is satisfied if one can find objects meeting the requirements (30/C130) expressed by the first term in the expression. If no such objects can be found, the requirements expressed can be satisfied by a set of objects meeting the requirements ϕ. Conditional expressions consist of a left parenthesis, followed by an expression, followed by a comma, followed by any number of additional expressions, each separated from its predecessors by a comma, and a final right parenthesis. They are evaluated by first trying to satisfy the first expression, then the second if the first cannot be satisfied, and so on to the end of the expression.

We shall want the expression ϕ to describe a transport aircraft whose troop-carrying capacity is a certain amount. That an object is a transport aircraft is determined by the fact that it exists in a tree of the form:



where the nodes under C130 and C124 denote tail numbers of aircraft. If we treat proper names as the names of the unit class that contains the object named, we might describe a set of tail numbers of transport aircraft as a set included in a set that is included in the set denoted by the string $x \ni x \subseteq^2$ TRANSPORT AIRCRAFT.

We will also need a way to indicate that the troop capacity of the members of this set must be the same as the troop capacity of 30 C130's. Troop capacity is a property of the set of aircraft; therefore, we can use the standard notation of the predicate calculus. Thus we might write:

TROOP CAPACITY (x) = TROOP CAPACITY (30/C130)

The full description of this part of the plan now has the form:*

$$((30/C120),\ (x \ni \subseteq^2 \text{ TRANSPORT AIRCRAFT and TROOP CAPACITY } (x)) = \text{TROOP CAPACITY } (30/C130))$$

A plan may also call for a change in condition. For example, it can require that an object located in one spot be moved to another. We can denote the change from a condition expressed by a string ϕ to the condition expressed by another string λ by the string $\phi \rightarrow \lambda$. When such a string appears in a plan it denotes a potential order. When the plan is activated, the string becomes an order. But similar strings in a description of the system's representation of the world can denote the execution of that order. When they appear in the rules that govern transformation in the representation, they can denote relations between such changes in condition.

Assignments

We have devised sufficient machinery to represent the requirements of plans. We now need to devise machinery to represent the assignment of resources to plans. We can denote resources by their names. If we have a squadron that has some name S, we can denote its assignment to a command C by the string $S^{\dagger} C$, and the assignment of that squadron to a plan P for some period or time interval t by $S^{\dagger} P(t)$. Underlying such statements are statements that indicate parts of objects and the command structure. A statement indicating parts of objects gives us the basis for identifying an aircraft assigned to a squadron S as also assigned to a command C, given only the string $S^{\dagger} C$.

---

*We use "and" and "or" to denote the truth-functional connectives in the obvious way.

A status of forces needs to tell us not only who is assigned to what command, but also information such as the location and condition of objects. We will associate a changeable property list to strings that denote objects. The fact that a C130 can carry 35,000 pounds of cargo does not belong on this list unless that property is being used: that is, unless the aircraft is loaded with that quantity of cargo. However, the location of a particular aircraft automatically belongs on the list.

Usually items on a property list will have time intervals over which the association holds. Let us denote the fact that a property list λ is associated with a particular string ϕ by writing $\phi\$\lambda$.

To represent the fact that a command C is to execute a plan P at time t we write:

$$C\$P(t) \tag{S-2}$$

If P contains a part of the form REQUIREMENTS ... (30/C130) ... REQUIREMENTS, the processing of S-2 requires that there be some string of the form $C^{\dagger}X$, where X is a string, in the representation of the situation such that X contains a part Y/C130, where Y denotes a number greater than or equal to 30. If this does not appear, the situation is now described completely as the result of applying given rules to strings, and we say that a problem has been recognized.

Problem Solution

To solve a problem the system has to do three things: (1) find the allowable alternatives, (2) select from the results of (1) some preferred alternative, and (3) issue the appropriate orders. These processes are relatively easy to describe in terms of the machinery already presented.

The alternatives available to a system are: (1) substitute the same kind of resource from some other command, (2) substitute an equivalent resource from the same command or from an alternative command, (3) permit slippage. The same resource exists in another command if, and only if, there exists in the system's representation of the world a string of the form $C^{\dagger}$ ... (X/C130) ... (where X is greater than or equal to 30). Substitution of resources is definable in terms of operations using conditional expressions. Slippage is permitted if none of these substitutions succeeds.

Selection of alternatives involves an application of values. It is difficult to determine exactly what these are in the case of a given system, but once a system's set of values is determined, we can assume the value of an alternative to be some function of its description. The value of this function is a number or a vector. For example, suppose that cost is the only value we are going to consider. Such a value can be computed for a given alternative ϕ. The result of that computation is a number X expressing the cost. This result can be represented by the string of the form ϕVX, where X denotes the cost. The choice of the alternative with the lowest cost can be readily described as an operation on a set of such strings.

VALIDATION

Under this contract we used the machinery of this appendix to describe the problems of two exercises of the 473L system.* We found this machinery adequate to predict the desired solution as well as alternative solutions. (Sometimes legitimate solutions appear that were not desired by the exercise designer.) We also found that the use of these models increased our understanding of the nature of the system.

APPLICATIONS

The deductive inference model demonstrates that the problem solving activities of command and control systems can be described in terms of string handling alone. Such a model has a number of applications in various phases of exercise and evaluation programs, including:

1. Exercise design, to help:

    a. Determine whether the information to be given to a system during an exercise is sufficient to describe the intended problem to the system.

---

*We have not included these descriptions in this appendix for three reasons: (1) they are long and hard to read without extensive explanation, (2) they are specific to the 473L system, and (3) they would require extensive changes to eliminate classified information.

b. Define the values of those parameters of the problem that should be controlled during the design phase. Examples of such parameters are the difficulty of recognizing that a problem exists, and the difficulty of solving a problem.

c. Determine whether an exercise plan really controls system behavior in the manner desired.

2. <u>Exercise monitoring and control</u>, to help:

   a. Organize control or contingency messages by classifying them according to their roles. This classification may be useful in planning manual or automatic exercise control.

   b. Organize monitoring by predicting the logically possible system behavior so that this behavior can be anticipated.

3. <u>Evaluation of exercise results</u>, to:

   a. Provide an ideal for comparison with observed performance.

   b. Help organize a description of these results for use in making recommendations.

4. <u>System evaluation and recommendations for system improvement</u>, to provide:

   a. A medium for the reallocation of system functions on the basis of exercise results, by providing a description of system functions relatively independent of implementation.

   b. A basis for user evaluation by generalizing exercise results to a large class of problems, given only the observations of system behavior in response to several problems.

5. <u>Design of command and control systems</u>, to provide:

   a. A medium for allocating functions among the various elements of a system.

   b. The basis for the organization of more flexible command and control systems in which the user defines his problem when it arises.

Since computers can carry out explicitly defined processes of the type we have described in this appendix, this model might also have some applications in:

6. The automation of some of the problem-solving activities of command and control systems.

7. The automation of exercise design, where one inverts the model. That is, the solution becomes the input, and the situation that leads to that solution is the output.

REFERENCES

1. J. H. Proctor, "Normative Exercising: An Analytical and Evaluative Aid in System Design," IEEE Trans. on Engineering Management, E10 (1963).

2. Peter Kugel and Martin F. Owens, "Some Techniques to Help Improve Methods for Exercising and Evaluating Command and Control Systems," ESD report ESD-TDR-64-195.

3. W. V. Quine, "Mathematical Logic" (Cambridge, Mass.: Harvard University Press, 1955).

4. T. E. Cheatham, Jr., "Translation of Query Language" (Appendix III Requirements for Support Programs in the 473L System), Tech/Ops Report TO-B 61-21.

5. J. W. Backus, "The Syntax and Semantics of the Proposed International Algebraic Language of the Zurich ACM-GAMM Conference," Proc. Internatl. Conf. on Information Processing (Paris, 1959).

APPENDIX II

INDUCTIVE INFERENCE MODEL

BACKGROUND

NATURE OF INDUCTIVE INFERENCE

Appendix I treats command and control systems as systems that deal, in certain ways, with deductive inferences. In this appendix we take the opposite view in a sense, and consider command and control systems as systems that make inductive inferences.

It is difficult to give a precise definition of the distinction between deductive and inductive inference, but it is not difficult to convey an intuitive understanding of this distinction. Roughly, deductive inference is the process of going from general assumptions to particular conclusions, while inductive inference is used in going from particular assumptions (or observations) to more general conclusions. This distinction may be illustrated schematically as follows:

$$\text{deduction} \downarrow \frac{\text{for all } x,\ \phi(x)}{\phi(a_1) \text{ and } \phi(a_2) \ldots \text{ and } \phi(a_n)} \uparrow \text{induction}$$

where x is a variable, but the a_i are the names of individual objects. Deductive inference appears to end up with no more than is given, helping to account for the certainty of the conclusions. Inductive inference in some sense amplifies the assumptions and thus appears to conclude things that do not follow necessarily. However, we do not intend to suggest that deductive inference is trivial. All mathematics is deductive. As we saw in Appendix I, when the deductions that can be made from a set of assumptions are sufficiently large, considerable ingenuity may be required to find a deduction that meets given conditions. At the moment, the problems that are involved in dealing precisely with inductive inference appear to be more basic than those involved in dealing with deductive inference. For this reason, the subject of this appendix has been less fully developed in the past, and our treatment is at a more fundamental level.

ROLE OF INDUCTIVE INFERENCE

Inductive inferences are made frequently in command and control systems; for example: (1) when an air defense system attempts to predict the intercept point for an aircraft that is zigzagging;* (2) when an intelligence system attempts to predict future enemy action from existing deployments; and (3) when a logistics system tries to plan its shipments on the basis of anticipated requirements.

Inductive inferences are characterized not by the fact that their conclusions are not stated with certainty, since probability theory yields such conclusions deductively, but rather by the fact that the conclusions appear to go beyond the given facts or premises. This appendix will show how inductive reasoning utilizes very general premises hidden in the past experience of the person making the inference. However, inductive inference deals only with relations between facts and does not consider values. (The way in which values are handled is the subject of Appendix III.) Inductive inference may allow us to predict that an opponent in a game of chess will get into a position that will cost him a knight if we make one move and into a position that will cost him a bishop if we make another. These are relations of facts. A judgment of value is required to choose which of these consequences we prefer.

ROLE OF RESULTS

The study of inductive inference is not as well developed as that of deductive inference, but it has evolved further than the theory of value judgments. In dealing with the deductive aspects of command and control systems, we were able to assume a set of basic axioms and to use the structure of those axioms as a background for emphasizing the peculiarities of a specific command and control system. In the treatment of values in Appendix III, we will observe that investigation is required to determine the nature of the basic objects to be dealt with. In our study of inductive inference we deal with the development of basic axioms into which the particular aspects of a command and control system might be mapped. (The mathematical content of this appendix will be developed into a paper for journal publication.)

---

*It is a deductive inference if the course of the aircraft is known.

PROBLEM TO BE SOLVED

The criteria one can use to distinguish between valid and invalid inductive inferences are not clear. This is not the case in deductive inference.* In order to decide whether a deductive inference of the form† ϕ implies λ is valid, it is sufficient to know that λ holds in every model in which ϕ holds, or more simply that ϕ, and not λ, is formally inconsistent.

The availability of a criterion for recognizing sound deductive inferences allows researchers in deductive logic to prove theorems and devise new tools. Unfortunately, such a criterion is not available for testing inductive inferences. It is easy to suggest that conclusions reached inductively should maximize simplicity, predictive power, or utility. It is another matter to make precise what these vaguely stated criteria mean. Currently, there is no general agreement on this matter. Certainly the criteria of deductive inference will not work. Given any finite collection of statements of the form $\phi_1(a_1)$, . . . , $\phi_n(a_{n+1})$, there are an infinite number of ways in which such a statement can be generalized, assuming that we have either an infinite number of available predicates or an infinite number of available names of individuals. In most formal languages we have both.

The problem of finding a justification for going from a finite number of statements about particulars to general statements about a larger class of particulars is an old one. Military systems depend on their ability to predict the future (future threats, future needs, future capabilities), and to do this they make inferences that go beyond the given facts. Exercises seek to develop and evaluate the ability of military systems to do this kind of thing.

The difficulties produced by the current lack of knowledge about the inductive process are particularly felt in exercising operators to perform inductive tasks. Since we do not know how experience is brought to bear on the performance of the

---

*This is not quite true, since the paradoxes of Cantor, Russell, and others, as well as the incompleteness theorems of Goedel, Rosser and others, show that this criterion will not work when pushed to extremes. However, it has been used in the development of the subject and continues to serve as a rule of thumb.

† ϕ and λ are variables ranging over statements. $\phi(a_i)$ and $\lambda(a_i)$ are statements about a_i .

task, we can only guess at how to sequence training exercises and the kind of experience that they should contain. The only way to evaluate the quality of a performance is to ask how well it would work in reality. Thus, the quality of our exercising becomes totally dependent on the quality of our simulation of enemy behavior. The main reason for the present study of inductive inference is to help overcome this dependence.

MODEL

BASIC IDEA UNDERLYING THE MODEL

In intelligence tests, frequently one is asked to continue a sequence of numbers, given the first few members. Thus, one may be asked to continue the sequence: 2, 4, 6, 8, ... The expected continuation is, of course, 10, 12, 14, . . . , but it could just as well have been 2, 4, 6, 8, 2, 4, 6, . . . , 8, 8, 8, 8, 8, . . . , 107, 67, 342, . . . , or anything else. The first of these is felt to be the most natural, based on one's education or other past experience. The validity of the IQ test depends on the fact that most people with the same background will tend both to have the same (relevant) background and to use it the same way. We shall be concerned with the question of how this might come about.

The example above captures the basic elements of what occurs when an operator tries to predict the future track of an aircraft on a radar screen. In theory, any prediction is possible. However, for all practical purposes, extremely improbable ones can be considered impossible (e.g., a 180° turn with zero radius), while others are very unlikely, given the past behavior of the track (plus the operator's experience). A good system may be judged by its ability to make such extrapolations. However, currently there is little that we can say about what is involved in making good extrapolations rather than bad ones.* We can recognize the former (they work) from the latter (they fail), and we can train operators by giving them more experience, although we have only the vaguest idea of how this works.

---

*Psychologists can describe some features of this process, but they cannot describe the mechanism accurately enough to be able either to predict or to simulate it.

To consider past experience mathematically, assume some sort of a mathematically definite system (e.g., a Turing machine) which receives information from a tape. (See Figure II-1.) This tape is divided into squares, and the machine

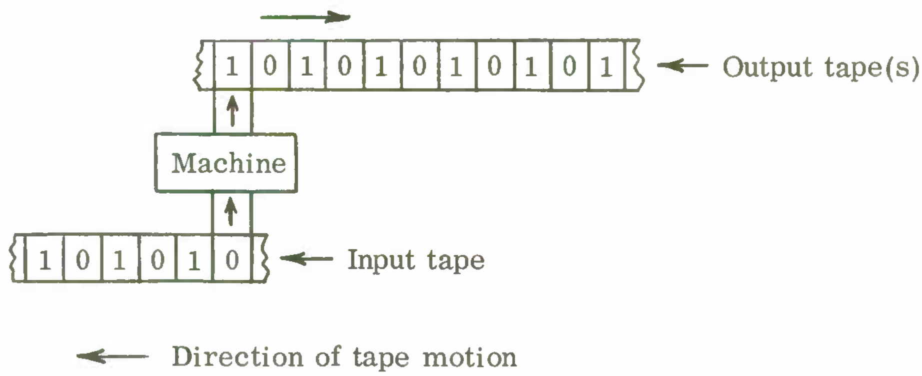


Figure II-1. Basic Model

reads the tape one square at a time in some fixed direction. The machine contains a definite program that outputs one or more tapes for every symbol that it reads. These tapes are interpreted as "predictions" of the remainder of the input tape. The remainder of this appendix will deal with (1) the relationship between the prior knowledge of this device concerning the nature of the tape it is reading and (2) the ability of the device to improve its performance in predicting the rest of this tape as it reads it. The former represents the underlying premises of induction and the latter, an operator's ability to improve his performance as the result of experience with the task. By representing only the bare skeleton common to all such situations, this model makes the process more susceptible to mathematical investigation. We will be concerned with situations where the machine neither has so much information that it is always right nor is so poorly informed that no program could possibly work.

Our machine is, of course, not an actual machine but a mathematical model of any such machine. It is much like the abstract automata that are used to represent the capabilities of computers. When such machines start to read a tape, they start off in some initial condition. The next condition of the machine and the tape(s) that it outputs are completely determined by the preceding state and symbol read by the machine.* We depart from most standard formulations only in distinguishing the source of inputs from the space used for "scratch" work during the computation. (These are both represented by the input tape in Turing's and most subsequent formulations of automata.) The following additional assumptions can be shown to result in no decrease of generality:

(1) The input tape rolls along in one direction and cannot be rewound.

(2) Each square contains one of two symbols, which we shall call 0 and 1.

LIMITATIONS OF PROBABILITY THEORY

First, let us see how far we can go with probability theory as it would usually be applied in this situation. Assume we are going to write a program for our machine that will simply get it to count appearances of symbols on the segment of the tape that it has already read and then predict the symbol that it has encountered most frequently in the past.† Where ties occur, we have it predict 0 and 1 alternatively. It is easy to see that such a device is not of much use in predicting tapes of the form:

10101010101010 ....

which we will henceforth denote by $\overline{10}$, the overline indicating that the sequence is repeated indefinitely. Here a human being would soon sense a pattern as he saw more and more of the tape, and he would begin predicting the next symbol without error. Our device, however, would be wrong 75% of the time (i.e., it would actually do worse than chance).

---

*The reader who is interested in a more complete and precise description of such machines is referred to Davis (Ref. 1).

†In our exposition we shall regularly switch between the prediction of the next symbol and the prediction of the whole tape. Since our devices are deterministic and we know their programs, these are equivalent.

A human being should start perfect prediction after he had read the fourth symbol (i.e., after he had read 1010, rather than after reading 101, which he might well continue as $\overline{101}$). Difficulties arise when we try to make explicit what is involved in the guessing of such patterns. This kind of task is fundamental in the inductive mode of command and control systems.

STRATEGIES

Let us now consider other procedures for predicting symbols on tapes, referring to such a procedure as a "strategy." A strategy can be considered to be a program built into the device. We shall want our programs to improve their behavior as more and more of the tape is read. In particular, we shall be concerned with determining the point at which the quality of prediction has reached an optimum. The occurrence of this point will represent the achievement of one particular goal of exercising: the training of an operator for a particular task.

A strategy is a procedure for storing information from the tape as it comes in, plus a procedure for outputting a prediction as a function of the stored information. An observer of such a machine may not know either its program or the conditions of its internal (or memory) tape. He may speak of the "behavior of the device" as the relationship between a segment of the tape of some fixed length n and the output produced immediately after reading this segment. This is the standard stimulus-response paradigm of the psychologist. However, it does not take into account that the length of the preceding segment that influences the behavior of such a device need not be determinable ahead of time. This may be arbitrarily long. It may even be a function of the entire contents of the tape or the entire experience of the organism. Psychologists have managed to avoid this additional complication and still get useful results because they deal with animals that may have limited capabilities. Human beings, however, appear to have greater capabilities.*

We shall be concerned with strategies that (1) change as more and more of the tape is read and (2) change for the better in that they predict more and more

---

*Devices of this stimulus-response type can be subjected to interesting mathematical treatments, but doing this is not our concern here. See Ref. 2 for a review.

symbols correctly. When such a device reading the tape $\overline{10}$ begins to predict 0 and 1 alternatively without exception (and in correct phase with the input), we will say that the device has "converged" because it has started giving the best possible guesses (which in this case are always right, but need not have this property).

We will be concerned with strategies that converge to the behavior that is the best possible, given whatever initial information about the tape is available to the system in the first place. We assume that the program used by such a machine is, in some sense, locked before the tape is read and that its program does not surreptitiously enter as data (as in Turing's universal machine). Thus, the initial information with which such a device can be provided can be looked at as information about the set from which the particular tape has been drawn.

To delineate the scope of the investigation, we shall consider two extreme cases: the best possible and the worst. Let us begin with the worst. We shall deal with a situation in which no strategy will improve one's chances of success. In order to do this we will have to define what we mean by "chances of success."

WORST CASE

Consider the case where we have no initial information about the set from which the tape has been drawn. To define the probability of drawing a given tape, we would like to say something like: "The population from which this tape has been drawn contains exactly one sample of each possible tape on 0 and 1, and the odds of drawing any particular tape are the same as any other." The trouble with this is that, since there is an infinite number of possible tapes, the odds of drawing any given tape are 0. Therefore, we must concern ourselves with sets of tapes and determine if there is any natural probability measure that we can assign to such sets. To say that sets of equal size should have equal probability will not work since no consistent measure that meets this requirement can be defined.*

---

*Proof: Suppose that this is possible. Define any function that divides the set of all tapes into two equivalent-size sets (e.g., all the tapes whose first symbol is 1 in one set, and those whose first symbol is 0 in the second). If sets of the same size must have the same probability, then both of these have the same probability as the other and as their sum (since they have the same cardinal number). But this is impossible since the probability of their sum must be 1.

One clue to the nature of a natural measure is provided by the fact that, although we have been dealing with tapes that are infinitely long, we are really basically interested in dealing with only finite segments of the tapes. The reason for letting them be infinitely long is simply that we are trying to deal with the situation where the length cannot be determined ahead of time. In general, we want to consider predictions for the next m squares, after n squares have been read. For $m = 1$, we would like the tape[*] X0 to have the same probability as the tape X1, since these are the only tapes of length $n + m$ beginning with X. By a similar argument, we would like the tapes X00, X01, X10, and X11 to have equal probabilities of occurring. Given this condition (extended for infinite sequences of finite tapes), there is a unique probability measure for the set of all tapes. The intuitive argument follows.

Observe that the set of all possible tapes can be set into a one-one correspondence with the real numbers in the interval $[0, 1]$ in a rather natural manner. This correspondence is established by letting the tape T correspond to the real number whose binary decimal[†] expansion is .T. This correspondence works almost everywhere, i.e., at all but an enumerably infinite number of points. The difficulty (which could be overcome) occurs with the numbers denoted by two distinct binary decimals: $.X0\bar{1}$ and $.X1\bar{0}$. These numbers (i.e., all numbers representable by terminating binary decimals) correspond to two distinct tapes. However, this makes no fundamental difference in our assigned measure.

If we identify tapes with the points on the real interval $[0, 1]$ that they can represent, we observe that if we already have read a sequence X on the tape, then the set of all continuations XT fall in the interval $[.X\bar{0}, .X\bar{1}]$. See Figure II-2. We note, also, that all tapes of the form X0T fall to the left of the midpoint of this interval, while all the tapes of the form X1T fall to the right. If we identify the

---

[*] We use X, Y, and Z to range over segments of tape. The tapes X0 and X1 are the two arbitrary tapes with the same initial segment, followed by a 0 in the first case and by a 1 in the second.

[†] This usage is sanctioned by Hardy and Wright (Ref. 3, p. 112) on the grounds that there is no viable alternative. T is a variable ranging over infinite tapes, and . T denotes the infinite decimal constructed by writing the contents of T after the decimal point.

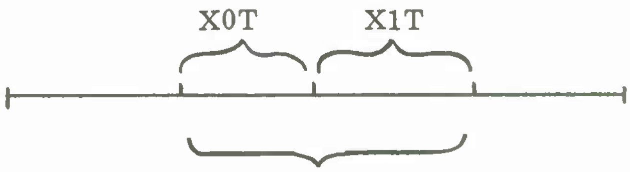


Figure II-2. Tape Continuations and the Corresponding Intervals on the Real Line

probability of Xa with the length of the interval associated with all real numbers denoted by numerals of the form .XaT, our requirement is always satisfied. Furthermore, it is easy to prove that no other measure meets it. Therefore, we are led to define (1) the probability of a given finite segment of tape as the measure of the set of all its continuations, (2) the conditional probability of a continuation of a given segment (P(Xa/X)) as the measure of all continuations of X divided by the measure of all continuations of Xa, and (3) the measure of any set of tapes as the measure of the corresponding set of points on the interval $[0, 1]$.

This enables us to state the following theorem:

> Theorem 1: Given the problem of devising a strategy for predicting symbols on a tape drawn from the set of all tapes on an n-letter* alphabet, the optimal strategy has the probability of success in predicting the next symbol of 1/n, and every strategy is optimal.

Proof: At a given point on a tape, any possible strategy will predict (by our definition of a strategy) a single sequence of length n as the next n symbols on the tape. By our definition of the probability of the continuation of a given tape, the probability of the tape XN (where N is a segment of length n), given X (= P(XN/X), equals $1/n^m$. But this is precisely the probability of every continuation and, therefore, the probability of success of any prediction whatever.

This theorem is trivial from a mathematical point of view. However, it is of interest because it says that unless one assumes something about the structure of

*In our case, n = 2.

the input beyond the alphabet that generated it,* one cannot make any distinctions between the goodness of alternative inductive strategies since they are all equally good. In other words, without additional assumptions about the structure of one's inputs beyond assumptions about the nature of their elements, inductive inference won't work. We need to look for additional assumptions which have sufficient generality to apply to any military situation. Such assumptions will have to be rather broad ones about the structure of the universe as it is sensed by both man and man-made sensors.

BEST CASE

The other extreme occurs if the set of tapes from which a tape is drawn is finite. Here we have the following:

> Theorem 2: Given the problem of devising a strategy for dealing with a tape drawn from a finite set of tapes, there always exists a strategy that converges to optimal prediction as soon as possible and whose probability of success at convergence is 1, provided that either: (a) each of these tapes can be generated recursively, (b) the device making the predictions has an infinitely large memory, or (c) prediction is required only for some fixed, given number of squares in the initial segment of any tape.

Proof: Case (a): To say that a tape (which contains an infinite number of occurrences of symbols) can be generated recursively is to say that there exists a finite program that generates it, one symbol at a time. Since it is clear† that any finite number of finite programs can be combined into a single program that generates the n^{th} symbol of each, followed by the $(n + 1)^{th}$ symbol of each, and so forth, let the predicting device contain such a program. At each point when n symbols have been read, let this program generate each of the tapes up to the first n symbols. If only one of these tapes matches the n symbols of the tape being read, output the rest of this tape. If more than one tape matches the symbols, let the

---

*The assumption that the tape contains distinct and distinguishable symbols is a strong assumption. However, it can be justified if one is dealing with a human inductor. There is some evidence that biological organisms have innate symbol discrimination mechanisms (see Lettvin et al., Ref. 4, Hubel and Wiesel, Ref. 5).

†For a proof, see Ref. 1, Theorem 2.1, p. 31.

predicted symbol in position m (for $m > n$) be the symbol that appears in the n^{th} position in the majority of the tapes that are continuations of the part read. If neither symbol is in the majority, write a 0 or a 1. It is easy to see that this program meets the conditions of the theorem.

Case (b): Let the device contain a copy of each tape, possibly with the squares of the tapes interspersed on a single tape (i.e., the m^{th} symbol of tape n on the $\left(n(m - 1) + m\right)^{th}$ square of the single memory tape). Proceed in a manner parallel to that of case (a), using look-up instead of generation.

Case (c): Proceed either as in case (a) or (b).

These two theorems set a bound on one part of the area under investigation. Theorem 1 says that if one has the most comprehensive* set of tapes from which to draw, then there is no interesting way to compare alternative strategies. Theorem 2 says that if one knows that one is dealing with a tape drawn from a finite set (and one knows what this set contains when one is designing one's strategy), then one can always devise (but not necessarily implement)† a strategy that is optimal in the following senses:

1. It converges to the best possible strategy, and this strategy has a probability of success of 1.
2. It converges to this strategy as soon as possible.
3. Its behavior, measured in terms of the probability of success of its predictions, is always the best possible, even prior to convergence.
4. It is possible to augment the strategy so it indicates when convergence has occurred.

This is the best we can do, and we will find increasingly general classes for which these various features have to be degraded.

---

*We cannot say "largest" because of Theorem 8, p. 60.

†The reason why implementation may not always be possible is that one can know the symbols on a tape that has been recursively generated without being able to figure out the program that will generate these symbols in this order.

PREDICTIONS WITH UNIQUE CONVERGENCE

In a limited number of cases one can devise strategies that converge to predictions that are optimal in the sense that (1) their odds of being successful are as high as possible and (2) there is no other strategy which has this property.

PERIODIC TAPES

We define a "periodic tape" as one that consists of the same sequence of zeros and ones repeated over and over again. Any periodic tape can be represented as $\overline{X}$. An example is the tape:

$$\underbrace{1101}\underbrace{1101}\underbrace{1101}\underbrace{1101}\underbrace{1101}\underbrace{1101}\underbrace{1101}\underbrace{1101} \ldots = \overline{1101}$$

We have the following theorem about prediction with periodic tapes:

> Theorem 3: Given a tape drawn from the set of all periodic tapes, there is a strategy that converges to the best possible strategy and does this as soon as possible. Upon convergence, the probability of success is 1. However, it is impossible to augment this strategy to have it indicate when convergence has occurred.

Proof: When the device has read a segment X, there are three possibilities:

(a) X = YY . . . Y, where YY . . . Y is a sequence of at least two occurrences of the segment Y.

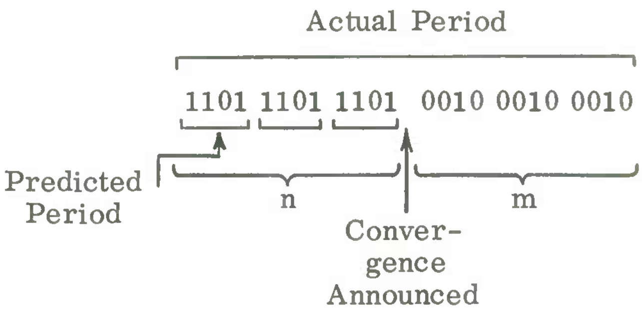
(b) X = YY . . . YZ such that Z is not empty and Y = ZZ′ . (That is, X consists of a repeated sequence of Y's, followed by some initial part of Y. E.g., $X = \underbrace{1101}_{Y}\underbrace{1011}_{Y}\underbrace{11}_{Z}$.)

(c) All other possibilities.

If either (a) or (b) holds, this can clearly be determined by a finite number of trials. The simplest way of making these trials is to test each initial sequence of the read portion X whose length is not more than 1/2 of the length of X to see if it meets the required conditions. Once this has been done, proceed as though X is the tape that is constructed out of the shortest possible Y. (If X consists of at least four repetitions of some Y, there will always be more than one way in which it could be represented as a sequence of segments.) If (c) holds, predict the first symbol of the tape, assuming that X is the initial segment of a tape of the form $\overline{X}$.

It is easy to see that this procedure will converge as soon as possible, since it will converge as soon as a single sample of the sequence of which the tape is generated has been read. Upon convergence, its probability of successful prediction will be 1, which is the highest possible.

However, we cannot augment this strategy to announce convergence when it has occurred, unless we are willing to settle for false announcements to an arbitrarily large amount. Suppose that such a strategy did exist, and take some periodic tape and run it through the device, implementing the strategy until convergence is announced. Assume this occurs after n squares have been read. Then consider a tape exactly like the original tape for the first n squares but different from it in the next m squares (i.e., it contains zero where the original contains one, and vice versa), and then is periodic with a period of length n + m. Example:



Assuming that the device in question takes seriously its own prediction of convergence (so that it maintains its guessing sequence unchanged), this device will be wrong with a probability that approaches m/(n + m). But since m can be as large as one likes, the probability of its failure can be as large as one likes.

This theorem has some philosophical interest in that it displays situations under which inductive inference is possible to the highest imaginable degree (it yields perfect predictions), but under which it is not possible for the device doing the inferring to determine whether the procedure that it is using is sound at any point in time. Indeed, as far as the device is concerned, the inference that it makes may not work again for any given finite interval. If this device is a human being, this means that it may not be correct again for the rest of its life. If the device is a command and control system, this means that it may not be correct again until it is too late. This is one aspect of what is often referred to as the "problem of induction," and Theorem 3 shows that it cannot be solved.

A periodic world is more than one can expect. Nevertheless, one might feel intuitively that the ability to deal with periodic tapes does lie at the base of inductive inference. Science, which is an example of the use of inductive inference, began with the prediction of periodic phenomena (e.g., the flooding of the Nile). The first sciences (e.g., astronomy) predicted phenomena that were perfectly periodic.

RATIONAL TAPES

It can be shown* that any periodic decimal (to any base) represents a rational number. Given the base, each such decimal represents a distinct rational number. However, not every rational number is represented by such a tape. Therefore, it is natural to consider generalizing the set from which a tape for prediction is to be drawn to include not only those tapes that represent rational numbers in the interval $[0, 1]$ that have periodic representations, but also tapes that represent any rational numbers in this interval. This generalization appears particularly relevant when one notes that the pure periodicity of the representation of a rational number is a function of the base selected. There is no particular reason why the base 2 should be the one that will eventually prove useful in applications. However, the notion of a rational number is independent of the base and is thus more fundamental.

Although not every rational number is periodic, every rational number is eventually periodic in the sense that at some point along its representation to any base it becomes periodic and remains so in its non-terminating representation.

Let us, therefore, consider the set of rational tapes, i.e., tapes that represent the names of rational numbers. This set consists of the periodic tapes and what might be called "impure periodic tapes." Impure periodic tapes are those

---

*See Ref. 3, pp. 111-112.

that can be represented as $Y\bar{X}$, but not as $\bar{X}$, nor as $Y'\bar{X}$ with Y' shorter than Y. Y is referred to as the "impurity." Example:

$$\underbrace{11010101110110110000}_{Y}\underbrace{\underbrace{1011}_{X}\underbrace{1011}_{X}\underbrace{1011}_{X}\ldots}_{\bar{X}}$$

The set of all tapes that are either periodic or impurely periodic will be called the class of "eventually periodic tapes." The facts that every rational number is representable by an eventually periodic tape (to any integral base) and that every such tape represents a rational number (though not a unique one) are proved by Hardy and Wright (Ref. 3, p. 111). It is easy to show that the results of Theorem 3 carry forward to the case of eventually periodic tapes. However, optimal convergence must be guaranteed somewhat differently.

Assume that we have a tape segment X. We look for the longest tail segment of the sequence X such that it matches an initial segment of X. (A tail segment is a segment at the end of the sequence.) This can be computed as follows: Denote the length of X by L(X). Consider the last $|L(X)/2|$ symbols of X.* If they are the same as the preceding $|L(X)/2|$ symbols of X, treat these as the period (unless an alternative period has been established by such a routine at an earlier stage in the processing of the tape that begins with X, in which case assume that period if this can be done consistently). If not, decrease the length of the tail segment being tested until a match of the appropriate type is found. The result of this procedure continues rational tape segments as follows:

	X	Continuation
a.	$\underbrace{1101}\underbrace{1101}\underbrace{1101}$	$\underbrace{1101}\underbrace{1101}\ \ldots$
b.	$\underbrace{11000}\underbrace{101}\underbrace{101}\underbrace{101}\underbrace{101}\underbrace{1}$	$\underbrace{01}\underbrace{101}\underbrace{101}\underbrace{1}\ \ldots$

*$|L(X)/2|$ denotes the integral part of L(X)/2.

But for the irrational tape[*] it yields:

c. 10110111011 1111111111 . . .

Examples (a) and (b) are in accord with intuition. However, for (c) we feel that the natural continuation would be

. . . 1101111101111110 . . .

We will return to this case later (p. 61). However, for rational tapes we have:

> Theorem 4: In the case of rational tapes or eventually periodic tapes, it is possible to devise a routine that predicts convergence as soon as possible and whose probability of success upon convergence is 1.

SEMIPERIODIC TAPES

One can do more to predict symbols on periodic tapes such as:

$$0011\,0011\,0011\,0011\,0011\,0011 \ldots \qquad (1)$$

than to say that at any point there is a 50-50 chance of either a 1 or a 0. Given a tape of this form, we expect to be able to do more than flip a coin to predict the next symbol, and clearly we can.

A periodic tape is one that consists of some finite sequence of 0's and 1's, repeated over and over again. (The sequence in (1) is 0011.) Suppose one has a tape constructed from more than one such sequence. (Call these component sequences $S_1, \ldots, S_s$.) If the order in which these subsequences appear is always the same, then the resulting tape is still periodic, although it has a longer period. However, consider the case where the order of sequences is determined randomly. For example, suppose we begin with the sequences 101 and 01110, which we will term "generating sequences." Further, assume that we toss black and white balls into a hat (in some ratio, R) and then start to write sequences out on the tape as follows. We draw balls from the hat (with replacement); every time we pick a white ball, we write 101; every time we pick a black one, we write 01110 on the tape.

---

[*]Theorem 136, Ref. 3.

We can speak of the tape generated this way as the "semiperiodic tape generated from the subsequences 101 and 01110 with a probability distribution of R." Where the distribution assigns roughly equal odds to both of these generating subsequences we might get a semiperiodic tape that looked like this:

$$10101110011101011011010111010101110011101010110 \ldots \qquad (2)$$

If one is going to try to predict occurrences of 0's and 1's on a tape like (2), it is clearly useful to know the kind of tape with which one is dealing, the subsequences from which it is generated, and the probabilities with which they occur. This knowledge could improve the quality of one's predictions. For example, if we assume only that (2) is generated from 0 and 1, we might eventually come to the conclusion that 1 appeared five times for every three occurrences of 0. Thus we might decide that the way to "predict" (2) is to predict 1 every time. This would give us a probability of success of 5/8. However, if we manage to figure out what the generating sequences are (we are not told this initially), we can improve our probability of success to 7/8. We would do this by noting when we were in the middle of a subsequence and by taking full advantage of this fact.

There are at least two interesting features in the situation discussed above. One feature is that the improvement of the predictions depends considerably on features of the basic sequences of which a tape is generated. Consider a series of sets of sequences $W_1, \ldots, W_i, \ldots$ where each set W_i in the series consists of two sequences:

$$W_i \begin{cases} \underbrace{00000000 \ldots\ldots\ldots\ldots\ldots 1} \\ \text{0 repeated i times} \\ \overbrace{00000000 \ldots\ldots\ldots\ldots\ldots 0}. \end{cases} \qquad (3)$$

If we consider tapes generated by the sequences in a W_i, then it is clear that the value of knowing the generating sequences (the members of W_i) of a semiperiodic tape goes down as i goes up.

A second feature of this situation is that an algorithm exists that will, given a sufficiently long piece of a semiperiodic tape, be able to determine the sequences from which the tape is generated and the probabilities that govern the relative frequencies of these occurrences, assuming that these are stationary. The existence of such an algorithm follows from the fact that a finite piece of tape can only be made up of a finite number of possible subsequences. Given any finite combination of "candidates" for the generating sequences, one can determine by algorithms whether assuming them to be the generating sequences of the tape would improve one's predictions. The evaluation algorithm would have built into it a sort of "nervousness" factor. This factor would determine how many times the given segment of the tape would be chopped up into smaller segments on which the same set of generating sequences would be tested. If this factor is 0, then the best guess at any point on the tape (say, the N^{th} tape square) is that the tape was generated by one subsequence of length N which had appeared exactly once.

The phrase "sufficiently long" at the beginning of the preceding paragraph implies that the distribution of subsequences in increasingly long segments approaches, to within some arbitrarily close limit, the distribution over the tape as a whole. However, the program of a predicting device cannot determine when it has a "sufficiently long" segment in hand. The device for predicting symbols on a periodic tape knew that its strategy would eventually lead to improvement in some measurable degree; here this is not the case. This is shown by the sets of sequences defined in (3), which gives us part 1 of the following theorem:

> Theorem 5: There exists a strategy that converges for any semiperiodic tape such that:
>
> 1. Given any ϵ, an infinite number of semiperiodic tapes exists for which this strategy will not improve the prediction by more than ϵ over the simplest possible strategy; i.e., the strategy that assumes that the tape is a semiperiodic tape generated by the shortest possible subsequences (0 and 1).
>
> 2. It is impossible to program a device to determine when it has finally reached the optimal strategy for a given tape, i.e., converged.

Difficulty 2 is the same as that for periodic tapes in Theorem 4. Difficulty 1, however, is a new "problem of induction." Roughly, it says that given any allowable margin of error, no matter how small, a device faced with a semiperiodic tape cannot be sure that the strategy it uses is any better than the simplest possible one within that margin of error.

A very natural way to find sequences that generate some semiperiodic tapes is to compute the autocorrelation function of the part of the tape already read. This function can be defined as follows: Let $a_i = 1$ if the i^{th} symbol on the tape is 1, and let $a_i = -1$ if the i^{th} symbol is a 0. Now, define the autocorrelation function $A(n)$ for a segment of length L as:

$$A(n) = \sum_{n=1}^{(i+n)=L/2} |a_i + a_{i+n}| \quad . \tag{4}$$

Given a sufficiently long slice of the tape, this function will peak at every n, where the length of a subsequence generating this tape is n. (But not every peak will be the length of a generating sequence.) In terms of this function, we can also deal with the case where the probabilities with which sequences occur are not independent. It is easy to see how this can be done, depending on the type of nonindependence. An alternative technique can be based on the observation that if sequences of length n have been used in generating tapes, this will show up in the transition probabilities of n^{th} order Markov Processes describing the tape. This is used in a related application on p. 66 below.

PREDICTIONS WITHOUT UNIQUE CONVERGENCE

In this section we deal with sets of tapes that cannot, in general, be handled by strategies that always converge to uniquely optimal predictions.

DEGREES OF GOODNESS OF RATIONAL PREDICTIONS

Assuming either periodicity or rationality, there is an infinite number of permitted continuations for any given segment of tape. Speed of convergence has dictated a choice of one of these, but it is not necessarily the most natural, and it leaves open the question of which continuation is second best.

Therefore, it is natural to extend the investigation to include the problem of ordering all possible continuations of a given segment (within the given assumptions) according to "goodness." Note that because of the word "goodness" this problem is not one that is susceptible to purely mathematical investigation.

An ordering according to goodness will be appropriate insofar as it approximates human behavior in this direction. For example, we will want an ordering that ranks 1010101 ... ahead of 11111111 ... as a continuation of 10101 ... (under the assumption that the tape considered is rational), not because there is any mathematical reason for this,* but rather because human beings prefer the former to the latter. What determines the adequacy of proposed ordering is not something intrinsic to it (although we would like the rule that defines it to be relatively simple), but the fact that it agrees with our intuitions, and eventually because it predicts choices that humans can be found (empirically) to make.

A useful ordering has been proposed by Dr. Franklin C. Brooks of Technical Operations Research. He suggests that tape continuations be ordered by the size of the denominator of the reduced fraction that they represent. The size of this denominator is clearly independent of the radix of the notation in which it is written, so that this ordering has the sort of universality over other bases that one would like. The fact that this ordering agrees as well as it does with our intuition is rather surprising.

The following examples illustrate where this happens. Suppose that we are given the initial sequence 110 and asked to continue it. According to the Brooks ordering, the continuations, in order of preference, are as follows (the underline marks the given sequence):

1. $11\bar{0} = 3/4 = \underline{110}0000\ldots$
2. $\overline{1100} = 4/5 = \underline{110}011001100\ldots$
3. $\overline{110} = 6/7 = \underline{110}110110110110\ldots$
4. $110\bar{1} = 7/8 = \underline{110}11111111111$

---

*There was such a reason when the tape was assumed to be periodic.

However, when one more symbol is provided as in 1101 the continuation as $\overline{110}$ is preferred, followed by $110\bar{1}$, and $\overline{1101}$.

THE BROOKS ORDERING

Another interesting case is the continuation of 011. The preferred continuation here is that the form of the tape is $0\bar{1}$ (= 1/2). The next in order, however, is not the expected 011, but rather $\overline{0110}$, and there is a sense in which this is particularly appealing. The second of these can be described in terms of a period of only length 2 and a rule that says switch symbols (0 for 1 and 1 for 0) every period.

Given a sequence D of 0's and 1's, let us define T(D) (the "transpose" of D) as the sequence that contains 0 where D contains 1 and 1 where D contains 0. It can be proved that if a rational number can be expressed in the form $.\overline{D\frown T(D)}$* it has a smaller denominator than a rational number that can be expressed in the form $.\overline{E}$, if the length of $D\frown T(D)$ is equal to the length of E.

The more rigorous statement of this is as follows:

Let $x = .\overline{D}$.

Lemma 1: x is representable as a fraction whose denominator is at most $2^{L(D)} - 1$, where L(D) is the length of the period D.

Proof:

$x = .\overline{D}$

$2^L x = D + .\overline{D} = D + x$ (since multiplying $.\overline{X}$ by the length of X has the effect of moving the decimal point over one period)

$(2^L - 1) = D$

$D/(2^L - 1) = x$ (QED)

Lemma 2: $1 - .\overline{X} = .\overline{T(X)}$

Proof:

$\overline{X} + \overline{T(X)} = \bar{1}$ (since if one were to write out the two left hand terms, one under the other, one would find one and only one occurrence of 1 in each column)

---

*The ligature denotes concatenation.

Since $.\bar{1} = 1$, $.\overline{T(X)} + \bar{X} = 1$. From this, the truth of the lemma follows algebraically.

Let $\bar{X} = \overline{\widehat{DT}(D)}$ for some D, and denote the length of X by L. L must be even so that L/2 is an integer.

$$2^{L/2}(.\bar{X}) = D + .\overline{T(X)}$$

$$= D + 1 - .\bar{X} \qquad \text{(by Lemma 2)}$$

$$(2^{L/2} + 1)(.\bar{X}) = D + 1$$

$$.X = (D + 1)/(2^{L/2} + 1)$$

Clearly $D \leq 2^{L/2}$. Thus we have shown that if $.\bar{X} = .\overline{\widehat{DT}(X)}$ it has a fractional representative whose denominator is less than that which we can find for any fraction of the form $.\bar{X}$ by Lemma 1, or:

> Theorem 6: Every rational number representable in the form $.\bar{X}$ to base 2 has a denominator $\leq 2^{L(X)}$, but if $\bar{X}$ is of the form $\overline{\widehat{YT}(Y)}$ it has a denominator $\leq 2^{L(X)/2} + 1$.

It is well known that $2^P - 1$ is prime if P is prime* for P = 2, 3, 5, 7, 13, 17, but that for P = 11 it is composite ($2^{11} - 1 = 89 \times 23$). This means that for all cases of period less than 9, for L odd, none of the fractional equivalents of $.\bar{X}$ can be reduced, although all those expansions of $.\bar{X}$ such that $X = \widehat{DT}(D)$ can be reduced. It is of interest to ask whether this condition of having a period that can be decomposed into $\widehat{DT}(D)$ is a necessary condition for having a denominator less than the upper bound set by Theorem 5 for periods of length less than 11 (since it is false for 11). An initial investigation of this question (which can be settled by computation) suggests that this is so, at least up to length 8.

Miller[6] has argued that psychological evidence suggests that the capacity of the human immediate memory appears to be about 7 bits. If one would like to argue that (1) the purpose of this memory is to order inputs according to the Brooks ordering and (2) the purpose of this ordering is to advance sequences whose periods are of the form $\widehat{XT}(X)$, then one might argue that the upper bound on the size of this

---

*Prime numbers representable in this manner are known as Mersenne primes.

memory is based, not on a neurological accident, but rather on the numerical fact that the procedure (of finding smallest denominators) fails to accomplish its purpose for larger numbers.

SIZE OF PREDICTABLE SETS OF TAPES

Theorems 3 and 4 concern periodic tapes and eventually periodic tapes and are special cases of a more general theorem, due to Dr. Brooks, which asserts:

> Theorem 7: Given a primitive recursive function $\phi(x)$ and a tape drawn from the set of tapes that this function enumerates, there exists a strategy that guarantees eventual convergence of the predictions of symbols on the tape.

Proof: Let $\phi(x)$ be such a function. Consider the tape generated by $\phi(x)$ for some given value of x. For a given tape segment of the form X, compute $\phi(1)$ until the first L(X) (binary) digits have been computed. If this matches X, predict that the tape contains the binary expansion of $\phi(1)$. If not, compute $\phi(2)$ for L(X) places in the binary expansion. If this process succeeds in matching the segment X, predict the expansion of $\phi(2)$. If not, continue computing $\phi(n)$, replacing n with n + 1 at each step. Since $\phi(x)$ is primitive recursive and since the tape T is drawn from the range of the binary decimal expansions of $\phi(x)$, this process must eventually exactly match T, and it does this at the first moment that there is no other tape earlier in the enumeration that matches the read segment exactly. This clearly is the earliest possible moment at which convergence can be guaranteed, although this procedure may fortuitiously lock on earlier.

PREDICTION WITH LIMITED ERROR

By our definition of a predicting device we have guaranteed that the number of distinct predictions be enumerable. Therefore, it is clear that no such device could predict a tape drawn from a non-denumerable set if we insisted both on convergence and on error-free prediction. However, if we relax the error-free condition we obtain a theorem that says that, however small a degree of error we desire, there is a non-denumerable set of tapes that can be predicted to within that degree of error.

To state such a theorem we have to define what we mean by "some given degree of error." Probably the most restrictive definition that we can give of the notion of a device that is correct at least m percent of the time is the following: A strategy for the prediction of symbols on a tape is correct at least m percent of the time on a given tape T if there exists some specifiable integer (n) such that for every segment of tape of length n, the method of prediction handles at least m percent of the symbols correctly.

Given this definition, we can now state:

> Theorem 8: Given any m less than 100, there exists a set of tapes such that (1) there exists a machine that predicts symbols on this set of tapes correctly at least m percent of the time and (2) this set is not enumerable.

Proof: Given any such m (which might even be transcendental), there is a number m' such that $m \leq m' < 100$ and which is representable by a terminating binary decimal. Consider the tape that consists of m' 1's followed either by a 0 or a 1, followed by another sequence of m' 1's, again followed by either a 0 or a 1. In this situation we can guarantee condition (1) immediately by simply predicting 1 each time. The fact that this set is not enumerable (condition (2)) is proved by simply crossing out all the sequences of m' guaranteed 1's. What remains is a set of arbitrary sequence of 0's and 1's. This defines a one-one correspondence with the real numbers in the interval $[0, 1]$ if we ignore the denumerable set of exceptions noted on p. 44 . This proves that our set of tapes has the cardinal number of the continuum and therefore is not enumerable, and the theorem is proved.

AN ALTERNATIVE ORDERING OF RATIONAL TAPES

The enumeration of rational tapes that we suggested in a preceding section is not the only reasonable one that can be conceived. In this section we will consider an alternative enumeration that is capable of generalization to the case of tapes drawn from the set of all rule-governed tapes (i.e., those covered by Theorem 7). In this case, again, we have a proof that convergence can be achieved (Theorem 7), and we want to order predictions by some sort of a priori device that corresponds to intuition.

An example may clarify the importance of this point. Consider the tape

$$0\overbrace{1}0\overbrace{11}0\overbrace{111}0\overbrace{1111}0\overbrace{11111}0 \ldots\ldots 0\overbrace{11 \ldots\ldots 11}^{n\ 1\text{'s}}0\overbrace{11 \ldots\ldots\ldots 11}^{(n\,+\,1)\ 1\text{'s}}0$$

This is not a rational tape (Theorem 136, Ref. 3), but intuitively we feel it is predictable.

If we know only that we are dealing with a tape that has been generated by a primitive recursive function, and we are given an initial segment of this tape such as

$$10110111011110111110,$$

there is an infinity of allowable continuations since there is clearly an infinity of strategies whose first outputs match the given segment and whose continuations are distinct from each other. We need to know why the continuation that we consider to be natural is the one that is preferred. We can do this within the terms of this appendix by giving an ordering of rule-governed tapes that matches empirical observations. To do this, we will generalize the notion used in this section to predict rational tapes.

The fundamental notion here is an old one that says that the complexity of a computer output can be measured in terms of the size of the smallest possible program that can generate it. In a somewhat different form, this argument has been developed by Solmonoff.[7]

Our enumeration of the rational (and later primitive recursive) tapes will depend on our enumeration of the programs used. This will require being more specific about the predicting device than we have been up to this point.

We assume that our machine (Figure II-3) contains (a) a copy of the segment of the input tape that it has already read; (b) a device that generates programs in a certain order; (c) a second device (the computer) that can be loaded with the program generated by the first device (the generator), and writes the results that the program produces on a segment of the scratch tape as long as the internal copy of

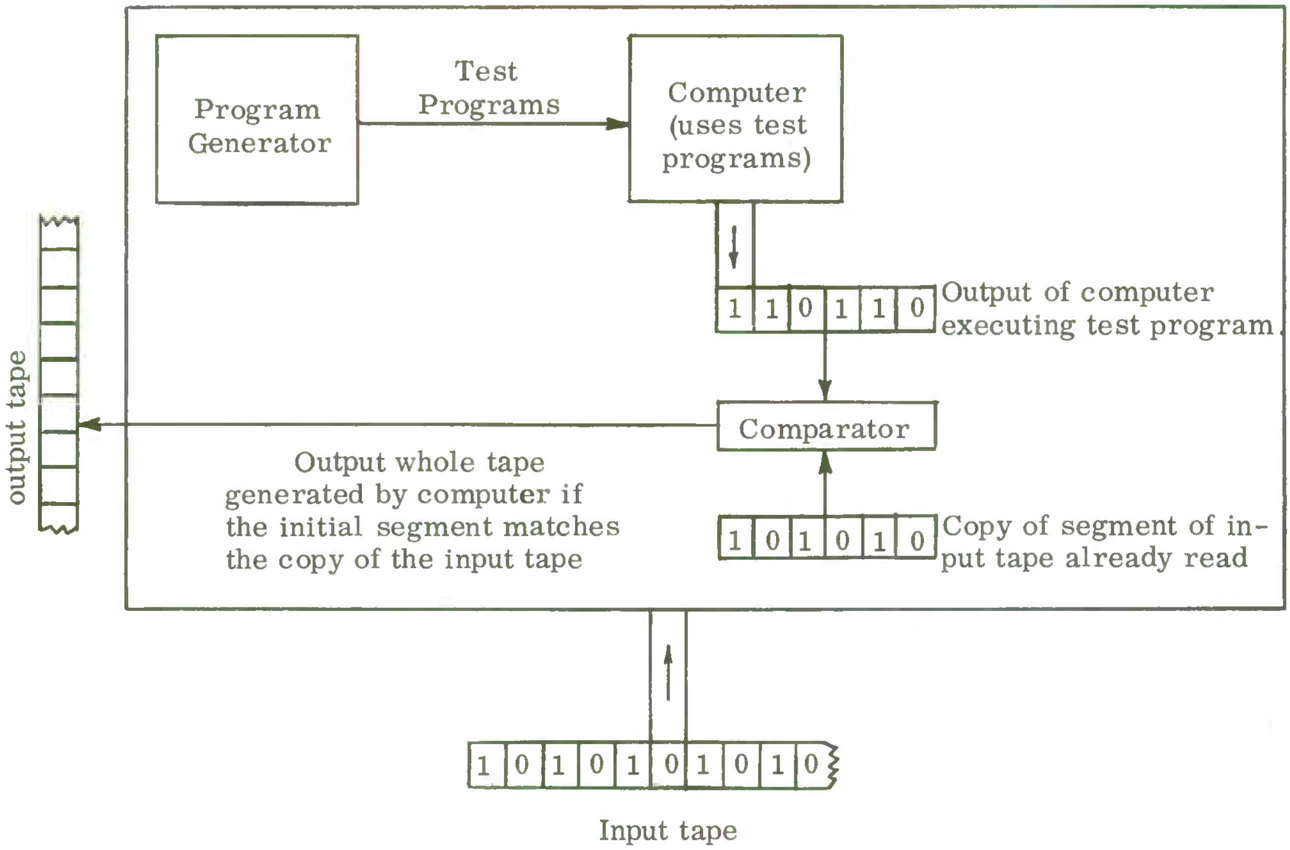


Figure II-3. Device for Enumerating Predictions

the input tape; (d) a comparator that compares the segment of the scratch tape produced at the end of this process with the copy of the input tape segment; and finally (e) a switch that is thrown when the comparator succeeds in matching the scratch tape and sample tape segments. The effect of throwing the switch is to have the program restart and to have the computer write its results on the output tape instead of the scratch tape. The kinds of predictions that such a device will make will depend on the kind of programs that it can generate and the order in which it generates these programs for testing.

Version 1: A program is a finite set of triples of the form $q_x S_y q_z$, where x and z range over the non-negative integers and S_y is either 0 or 1. These triples are interpreted by the computer as follows: The computer begins in a state q_0.

A triple of the preceding form is executed if the machine is in state q_x. In executing a triple $q_x S_y q_z$, the machine prints the symbol S_y on its scratch or output tape and shifts to state q_z. This process continues (the machine is now in q_z) until the machine gets into a state such that no such state name is a first member of a program triple. At this point it halts. The basic notion of our enumeration is to enumerate these programs in order of their length. Because of the way our total machine works, we do not have to consider either programs that can be shortened (i.e., which generate the same output as a shorter program) or programs that do not generate infinite outputs. Furthermore, we need not consider separately those programs that are similar to each other, beyond renaming their q_x's. We will want to make sure that we consider only programs whose behavior is fully determined.

These conditions lead to the following postulates to limit the programs we consider, where P is an arbitrary program:

1. The computer is deterministic: $q_x S_y q_z \epsilon P$ and $q_{x'} S_{y'} q_{z'} \epsilon P$ $\rightarrow (x \neq x' \rightleftharpoons y \neq y'$ and $z \neq z')$.

2. Every instruction goes somewhere: $q_x S_y q_z \epsilon P$ $\rightarrow (\exists T)\ (T = q_{x'} S_{y'} q_{z'}$ and $x' = z)$.

3. Every instruction is used:

   Define "successor" as follows: q_z is the successor of q_x if, and only if there is a $q_x S_y q_z \epsilon P$. Define $\dagger(q_x, P)$ as the set of all states of P that can be reached by reiteration of the successor operation. The postulate now reads: $q_x S_y q_z \epsilon P \rightarrow q_x \epsilon \dagger(q_0, P)$.

4. Programs that are just reletterings of each other (aside from reletterings of the S_y) are considered equivalent. That is, two programs are equivalent to each other if there exists a one-one mapping from the states of one to the states of the other that maintains the relationships within triples. (If the image of x is denoted by f(x), then $(q_x S_y q_z \epsilon P \rightleftharpoons f(q_x)\ S_y\ f(q_z) \epsilon P') \rightarrow P = P'$.)

We enumerate such programs in order of length (number of instructions), and within length to the sum of the sizes of the ancestrals* of the successor relation of all the states within them. We observe that:

1. This always defines a prediction for any tape, or a sequence of predictions if one lets the machine keep printing output tapes as it continues down the enumeration.

2. Such programs generate all and only binary decimal representations of rational numbers in the interval $[0, 1]$.

3. This ordering differs from the enumeration of the rationals in order of the size of the denominator. (See example (a) below.)

4. This is a reasonable enumeration as a predictor of what we consider to be good continuations.

Examples:

(a) 1011 is continued 011011011011011 ... rather than 1011101110111011 .... (Compare this with the prediction of the alternative on p. 56.)

(b) 11010 is continued 110101101011010 ... rather than 110100000000000 ...

(c) The sequence of n 1's followed by a 0, followed by m 1's is continued periodically rather than with an infinity of 1's, providing m is sufficiently small.

GENERALIZATION TO RULE-GENERATED TAPES

A further advantage of the preceding enumeration is that it can be generalized as follows:

Version 2: Let the programs be sets of quadruples of the form $q_x S_y^t S_z^{t'} q_w$, where x and w are non-negative integers, t and t′ are o or s (and indicate which of

---

*Given a relation R(x, y) and a set S, the ancestral of R(x, y) in S is the set of all z such that there are z^i's in S for which $R(x, z^1), \ldots, R(z^{i-1}, z^i)$ and such that $z^i = z$.

two tapes the symbol is to be read from or written on), S_y is either 1 or 0 or B (for "blank"), and S_z is either 1, 0, B, R, or L. These quadruples are interpreted as follows: Again an initial state q_0 is selected. When a machine is in state q_x <u>and</u> it reads the symbol S_y on the tape t, then it

1. Prints 0, 1, or B on the output tape if $S_z^{t'}$ is 0^o, 1^o, or B^o, respectively.

2. Prints 0, 1, or B on the scratch tape if $S_z^{t'}$ is 0^s, 1^s, or B^s, respectively.

3. Moves the scratch tape one square to the left or right if $S_z^{t'}$ is L^s or R^s, respectively.

4. Moves the output tape one square to the left or right if $S_z^{t'}$ is L^o or R^o, respectively.

Our machine now has two scratch tapes, one of which serves as a potential output tape (if it matches the sample); the other is used for a temporary memory.

The main motive behind this construction is suggested by the following program:

$$\left.\begin{array}{l} q_0\ B^s\ 1^s\ q_0 \\ q_0\ 1^s\ R^s\ q_1 \end{array}\right\} \quad \text{prepare the scratch tape (S)}$$

$$\left.\begin{array}{l} q_1\ B^o\ 0^o\ q_1 \\ q_1\ 0^o\ R^o\ q_2 \end{array}\right\} \quad \text{write 0 on the output tape}$$

$$\left.\begin{array}{l} q_2\ B^s\ L^s\ q_3 \\ q_3\ 1^s\ 1^o\ q_4 \end{array}\right\} \quad \text{copy the 1's on S onto the output tape}$$

$$\left.\begin{array}{l} q_4\ 1^s\ R^s\ q_3 \\ q_4\ B^s\ 1^s\ q_5 \end{array}\right\} \quad \text{add a 1 to the string on the scratch tape}$$

$$\left.\begin{array}{l} q_5\ 1^s\ R^s\ q_5 \\ q_5\ B^s\ B^s\ q_1 \end{array}\right\} \quad \text{rewind the scratch tape and go back to } q_1$$

This generates the tape 010110111011110111110 . . . .

A program to generate a periodic tape of length L has to write the numerals of the period onto the scratch tape, requiring L instructions; rewind the scratch tape (which we shall assume to be endowed with an end marker), requiring two instructions; and copy the contents of the scratch tape, requiring three instructions. Thus, the total requirement of the program is that it take L + 5 instructions, although there will be cases where this can be decreased. Since the preceding program takes 10 instructions, our routine suggests that after five symbols have been read (i.e., 01011), the periodic continuations 010110101101011 ... and the original 0111011110111110 ... are both equally likely since both require programs of length 10. Once one more symbol has been read and the segment 010110 is in hand, the correct continuation is slightly preferred. This seems to coincide with intuition.

SPECIAL CASES

NOISY INPUTS

If we are to be concerned with real devices or systems, some attention should be paid to the fact that it is unreasonable to expect inputs to represent exactly what is on the tape. In other words, there will be noise. In spite of this, we would probably prefer to have the tape:

$$\underset{\text{noise}}{101010101010101010101010100101010101010101010101010\ldots}$$

continued as $\overline{10}$ rather than by 100, which repeats the noise periodically.

We define a "Markov Process of order n" as a procedure that predicts on the basis of expected transition probabilities, where probabilities are based on n-grams (sequences of 0 and 1 of length n). Given the case of the preceding sequence, to predict $\overline{10}$ this device might use a second order Markov Process defined by the following transition probabilities derived from the given segment:

	00	01	10	11
00	.0	0	.02	0
01	0	.98	0	0
10	0	.02	.98	0
11	0	0	0	0

It is simple to see and to prove that such devices will eventually predict occurrences of 0 and 1 on any perfectly periodic tapes and will do creditable jobs on noisy periodic tapes.

To handle noise on rule-governed tapes, instead of the usual type of Markov Process consider a slightly more complex version. Assume that we have some finite set of rules (or operations) $R_1, \ldots, R_n$. Assume that in some manner we have also found a number of subsequences $S_1, \ldots, S_m$. Let us denote the result of applying rule R_i to some one of these subsequences S_j by $R_i(S_j)$. If we assume that this results in some other sequence, then one of the rules might be applied to this result, and so forth. Consider the set of all such results generated by applying a given rule R_i to any rule-generated sequence and denote it by $R_i(S)$. Now consider the transition matrix of the following form:

$$\begin{array}{c|ccc} & S_1 \text{-----} S_i & R_1 \text{-----} R_j & \\ \hline R_1 & P_{1,1} & \text{--------------} & P_{1,(i+j)} \\ \vdots & \vdots & & \\ R_j & P_{1,j} & & \end{array} \tag{5}$$

such that

$$\sum^{m=j} \sum^{n=(i+j)} P_{n,m} \geqq 1$$

The > 1 occurs if some S_a is also $R_b(S_c)$ or if some $R_a(x) = R_b(y)$ for $a \neq b$. Such a matrix tabulates the probabilities that if a given sequence appears on the tape, and we recognize it either as a basic sequence or as one generated by some operation R_i, the given sequence will be followed immediately by the result of some application of the rule R_j to the resulting subsequence.

Note that if we take R_1 and R_2 to be the rules "write a 0 after" and "write a 1 after" with $S_1 = 0$ and $S_2 = 1$, then we have a special case that exactly covers semi-periodic tapes as a special case of the rule-generated tapes. In this interpretation, the sequence 01110 could be represented by the row entry in a matrix of the form of (5), which looks like this:

$$R_1 \left\{R_2\left\{R_2(R_2(S_1))\right\}\right\}$$

The rule generated tape 010110111011110 ... can now be represented by the simple transition matrix:

$$\begin{array}{c|c} & S \\ \hline R_1(S) & 1.0 \end{array} \qquad (6)$$

This says that each subsequence is the result of writing a 1 after the string segment that preceded it. It is also easy to see that small amounts of noise would not disturb this procedure.

Note that the sequence 1, 2, 3, ... is not of such a simple form. For this sequence, one needs a number of generating subsequences (the digits 0 thru 9) and a number of operations ("write a 0 after S," "write a 1 after S," ...). Things get even worse if one tries to formalize the rules for saying the names of numbers.

Although it is clearly simpler to use a unary representation or description of N, there are extremely good reasons why neither human beings nor computers do so simple a thing. It would take human beings too long to say "zero, one, one ... , one" with n repetitions of one to name the number n. (It would also introduces errors, if it was understandable at all.) The addition of new symbols and new processing rules can have advantages for devices that need to concern themselves with limitations of time and space. It is easy to conceive of hardware configurations that make notation to a large base more efficient, even though its utilization requires additional rules of operation.

Until recently, studies of computation from a mathematical or abstract point of view have tended to disregard such limitations. If one does disregard them (and there are good reasons for doing it), one has no really sound way of distinguishing sets of operations as being more or less complex than others.* Thus, automata

---

*Except, of course, the special case where one is properly included in the other.

theory is full of equivalence results showing that some sets of operations are exactly the same as some other sets. As long as this situation prevails, it is difficult to apply the rule matrices discussed above because the choice of row and column headings is arbitrary.*

Recently, however, there has been an increased interest in distinguishing automata in ways that are closer approximations to criteria that approach the nature of hardware limitations. Among these are the approaches of Myhill,[8] distinguishing operations (functions) in terms of how much space is required for their computation, and the work of Ritchie,[9] Yamada,[10] and Hartmanis (unpublished), distinguishing operations (functions) in terms of how long it takes to compute them. All these results emphasize the computation of mathematical functions. If one is concerned with more general functions, the classification of operations might be based on somewhat different notions. Some ideas about these notions are the subject of the next section.

HARDWARE LIMITATIONS

In the preceding section, we suggested that rule-application matrices might be useful for predicting symbols on the class of rule-generated tapes. However, we noted that it was not easy to determine a basic set of rules to define the initial values of the rows and columns of such matrices. Specifically, it is not as easy to find such basic rules as it was to find a suitable basic set of symbols for predicting symbols on semiperiodic tapes, which are special cases of the set of rule-generated tapes.

The purpose of this section is to discuss some features of an approach to the development of such matrices by studying the basic operations used by information-processing automata (computers, Turing machines, and the like). The purpose of such an investigation is to derive a set of elementary or basic operations, each of which would be independent of the others and would be minimal. In terms of these elementary operations, more complex rules could then be built up over time by a

---

*This is not true for the choice of basic symbols because it is clear that the simplest universal set contains exactly two symbols. It is not equally clear, however, which set of rules is the simplest set of universal rules.

device that was attempting to predict symbols on rule-generated tapes — much as more complex symbols were built up over time from the original symbols (0 and 1) used in initiating the types of devices for dealing with the semiperiodic tapes described in previous sections.

In trying to develop a set of such elements, it may be useful to try to characterize computation or information processing in a highly formal manner, in much the way that Hilbert characterized Euclidean geometry. Where Euclid relied on our spatial intuition, Hilbert tried to make explicit the features upon which this intuition depended. Most current-day formulations of the theory of computation depend on the modern-day version of our spatial intuition, which might be called our machine intuition. This dependence is shown by the regularity with which pictorial metaphors appear in the definitions of such devices. (We used such pictorial notions in Figure II-1.)

As long as one sticks to pictorial notions, it is not easy to compare or to keep track of differences of organization. But it is precisely these differences of organization that the state of a rule-application matrix is going to influence at some given moment. Therefore, the development of a theory capable of dealing with rules in an explicit manner appears to be a prerequisite of the development of rule-application matrices.

Once there is a rigorous theory of automata in which the organization of the automaton is itself a variable, capable of being studied within the theory, one can deal with the structure of the automaton as one of the variables that one attempts to get to converge over time. In other words, just as we have attempted to have automata converge in their choice of basic sequences of symbols over time in the preceding sections, the type of theory suggested here might provide the raw materials for a theory dealing with the development of complex operations over time.

If we consider an automaton, we can distinguish two types of elements that we shall call the static and the dynamic elements. The static elements are those involved in describing the state of the device at any given moment of time t, and the dynamic elements are those involved in describing the rules according to which the automaton changes its states from one moment t to the next (t + 1).

Another way of distinguishing between these two types of elements is to observe that the static elements are finite but unlimited in size (for a universal machine), while the dynamic elements are always fixed in size. Thus, the static elements in a computer are represented by its tapes; another tape can always be mounted when the one that is mounted has been used up. On the other hand, the dynamic elements are represented by the machinery associated with the registers and used to change the static elements. These registers are fixed in size, and there is an initially-fixed limit on what they can do in any given machine cycle.

Static Elements*

The status of the device at any given moment (the static elements) is the only record the device has of what has happened to it in the past. Computers and most abstract models of them store these data (1) by placing particular symbols selected from some alphabet in storage or (2) by arranging these selected symbols in a particular order. These modes of storage are interchangeable.

This interchangeability is a fact that the use of computers has made familiar for at least one special case. Storage in ordinary written English is done in terms of symbols chosen from a twenty-six letter alphabet plus some other symbols used for spacing and punctuation. These symbols are then arranged in serial order. However, a computer is capable of storing only sequences of symbols drawn from a two-member alphabet. In spite of this, by using appropriate encoding that requires longer sequences but fewer basic symbols, computers are capable of storing any sequence of letters in the larger English alphabet.

There are two ways in which the information encoded by this dual method is used. The location of the symbol sequences may be relative (e.g., the third occurrence of 1 in the binary representation of 11 (1101) is to the right of the second occurrence of 1) or it may be absolute (e.g., the symbol 1011 is in the storage register). Fixed locations (i.e., registers) are established prior to the beginning of a computation so that their size and number are fixed for the duration of a computation. Thus changes of the state of the automaton, which are unlimited in extent,

---

*The effect of limiting the static elements of devices to predict semiperiodic tapes has been treated in an elementary way by Cetlin.[11]

are always determined by steps, and the size of each step is fixed at the start of the computation. The size of these fixed elements, measured by the state-symbol product, is one of the limiting organizational features of such devices. It helps to determine which rules are simple and which are complex.

A variable in the organization of the static elements of data processing devices, which helps to determine the simplicity of rules, is the organization of the relatively located elements. In the ordinary computer, such organization is based on a fixed word-size, or in the organization of the memory into various other types of blocks. An example of the difference such a factor makes is provided by certain operations that are simple on a fixed word-size machine but are difficult on a variable word-size machine, and vice versa.

These examples suggest how the organization of the static elements of a computer (or other information-processing devices) can influence what the elements of computations are on such machines and what combinations of them are simpler than others. Thus, they are basic to the determination of elements of rule-application matrices.

Dynamic Elements

The dynamic elements operate on the static elements. They evaluate functions from the state S(t) of the static elements at some time t to a state S(t + 1) of the static elements at a later time (t + 1). Such functions operate in terms of the registers and can change only fixed amounts of the static elements at a time. These amounts and the sorts of changes that can be produced in a cycle are basic variables. These variables also determine what constitutes a simple rule.

Considerations of the variables in such static and dynamic elements might lead to ways for defining the elements of rule-transition matrices and for operating on such matrices to define new operations that are useful for approaching convergence. Preliminary analysis suggests that this is true. Further analysis might help to develop the basic elements of a theory of predicting rule-governed tapes. A study of such elements might serve much the same purpose in an understanding of the relationship between rule-governed tapes and rule-executing automata as a study of the parts of chemical elements does in studying the relationships between the chemical elements.

APPLICATIONS

In this appendix we have described the elements of a theory of inductive inference that treats the relationship between the type of "world" with which a method of induction can deal and the nature of that method. This theory can be applied both to the exercise and evaluation of command and control systems and to the training of their personnel. The usual procedure for evaluating an operator who works in an inductive mode is to see whether he succeeds in actual practice. Since some of the most important command and control system functions (i.e., those for the case of a general war) are going to be used only once, one is forced to test the success of an operator on a simulation of the anticipated situation. Therefore, one's test is only as good as the simulation. There is no simple way of measuring an operator's capabilities on the basis of such tests.

Using the theory described in this appendix as a basis, one might be able to describe an operator's capabilities in terms of the set of possibilities (tapes) that he was capable of dealing with correctly. Training would consist of providing experience with ever broader sets of possibilities until the desired set had been obtained.

The use of this theory in the exercising of a command and control system is parallel to the use in training individual operators, but with the particular difference that its use might help provide design guidance. An exercise could be used to determine what set of possibilities a given system was capable of handling. It could do this by sampling the various sets of possibilities, described in the spirit of the theory of this appendix, and testing the system's capabilities with these samples. Given such a description, one could then go to the user and consider how this set of capabilities might be extended in subsequent versions of the same system.

The theory discussed in this appendix might also be used in the construction of devices which automatically perform the type of induction described in the theory. Such devices might have applications in the general area of pattern recognition.

REFERENCES

1. M. Davis, "Computability and Unsolvability" (New York, N.Y.: McGraw-Hill, 1958).

2. Saul Sternberg, "Stochastic Learning Theory," Handbook of Mathematical Psychology, vol. II., ed. by R. D. Luce, R. Bush, and E. Galanter (New York, N.Y.: John Wiley and Sons, 1963).

3. G. H. Hardy and E. M. Wright, "The Theory of Numbers" (New York, N.Y.: Oxford University Press, 2d ed., 1963).

4. J. Y. Lettvin and others, "What the Frog's Eye Tells the Frog's Brain," Proc. IRE 47, 1940-1951 (1959).

5. D. H. Hubel and T. N. Wiesel, "Receptive Fields, Binocular Interaction and Functional Architecture in the Cat's Visual Cortex," Journal of Physiology 160, 106-154 (1962).

6. G. Miller, "The Magical Number Seven, Plus or Minus Two: Some Limits on Our Capacity for Processing Information," Psychological Review 63, 80-97 (1956).

7. R. J. Solmonoff, "A Formal Theory of Inductive Inference Part I," Information and Control 7, 1-22 (1963).

8. J. Myhill, "Linear Bounded Automata," WADD Technical Note 60-165 (1960).

9. R. W. Ritchie, "Classes of Predictably Computable Functions," Trans. American Mathematics Society 106, 139-173 (1963).

10. H. Yamada, "Counting by a Class of Growing Automata," Doctoral Dissertation, University of Pennsylvania (1960).

11. M. L. Cetlin, "Finite Automata and the Simulation of the Simplest Forms of Behavior" (in Russian), Uspehi Mat. Nauk 18, N. 4, 3-28 (1963).

APPENDIX III

VALUES AND DECISION MAKING IN COMMAND AND CONTROL SYSTEMS

INTRODUCTION

Large quantities of information have been gathered on the performance of military command and control systems. Data about mean system-response time, number of aircraft detected, and average system downtime abound. Yet few data exist on the most important operational component of a command and control system — the functioning of the military decision maker. Most of our military command and control systems are manned in shifts around the clock. The decision makers change constantly. Can we rely on these operators to make the "right" decisions? Is there any continuity in the decision-making function? Can we predict how operators will make decisions? Can we select those operators who will make decisions the "right" way?

The answers to these questions are critical to the operation of a command and control system. The ability of the main decision makers to implement their decisions determines the success or failure of the system when tension is high, information is scanty, and time is short.

This appendix discusses our investigation of one aspect of the human element in the command and control situation, the relation of the decision maker's value judgments to his decision-making behavior. The discussion is based on a study designed (1) to examine the feasibility of establishing, for members of an experimental population, "value" indexes based on abstract value judgments, and then (2) to relate these to decisions as manifested by the subjects' responses to items of a decision questionnaire. While we have no firm answers to the questions we have posed, we believe that we have a line of investigation that may lead to these answers. However, owing to the limited effort expended on the work described in this appendix, our results are limited, both in generality and validity.

THE PROBLEM

The response of a command and control system should be uniform and should be "right" with respect to the values held by the higher authority whom the system serves, regardless of the composition of the duty team. For example, the 473L system serves the Chief of Staff, USAF. If it were his policy (unwritten) to change DEFCON levels for certain commands at the slightest sign of trouble, then this should also be the policy of all duty officers who might man the decision desks at any time.

It is our view that if all decision makers of the system attach the same values (or utilities) to the possible outcomes of a decision, this uniformity will contribute to the desired consistency in the system's decision process.

Elements such as personality type, intelligence, amount of training received, amount of pertinent current knowledge, and belief in available data affect the individual operator's decisions. However, we intend to concentrate on the value area in this appendix. Specifically, we are interested in the values attached by the system operators (decision makers) to the possible outcomes or results of a decision. Furthermore, if we can demonstrate that a relationship exists between a person's value spectrum and the way in which he renders decisions, then we should be able to select personnel for command and control systems who possess the appropriate decision-making attributes. Once the individual value items appropriate in a particular system have been isolated, it should be possible to train operators in these values and to exercise them further in order to engrain the values.

Because we believe that a positive relationship does exist between personal value and decision making, we undertook a small pilot study in which we attempted to develop the tools necessary for a major study of the value/decision relationship.

THE STUDY

Our first task was a search for the value factors related to decision making. The value factors chosen were selected intuitively after a literature search revealed no clues to research in this area. Introspection by the author and discussions with colleagues familiar with command and control system technology resulted in the choice of thirteen value factors that were thought to be relatively independent and influential in command and control decision making. These factors are: Time, Money, Human Life, Territory, Goodwill, Political Advantage, Intelligence Data, Weapon Stockpile, Weapon R and D, Industrial Capacity, Technological Superiority, First Strike Capability, and Second Strike Capability.*

It was felt that the values people would assign to each of these thirteen factors, which may actually be thought of as resources, would vary under different world conditions. Therefore, we devised a series of nine situations involving the United States in progressively more severe (or dangerous) conditions. These situations are: Arms Reduction and Control, Intensified Cold War, Allies Threatened, U.S. Threatened, U.S. Severely Provoked, Allies Attacked, U.S. Attacked, U.S. in Limited War, and U.S. in General War. Again, the choice of these world conditions resulted from cogitation and consultation and indeed were not, as it turned out, in proper sequence. We had rated "U.S. in Limited War" a much more severe condition than did most of the subjects. This will be discussed in more detail later.

Next, we recruited a set of subjects to serve in our pilot experiment. A group of subjects composed of military decision makers would have imparted greater face validity to the study. Since military subjects were not available, a group of twenty-four scientists from Technical Operations Research was chosen.

A questionnaire (pp. 91-93) was constructed to investigate the subjects' value structure with respect to the thirteen value categories and under the nine different world situations. This questionnaire required the subject to allocate 100 ults (arbitrary units of value) among the thirteen resources for each situation. The instructions requested the subject to assume the role of Chairman of the Joint Chiefs of Staff, who has been directed by the President to indicate to him how the country's resources are to be invested so that maximum preparedness will result. The subject was asked to accomplish the value distribution twice. First, he was asked to allocate the ults in such a manner that the attribution indicated a value judgment of the relative worth of each resource as an asset of the United States. In the second ult allocation, however, the criterion was the relative worth of the

---

*This was an exploratory study. It is recognized that a systematic development of measuring instruments would involve scale construction, validity, and reliability analyses. Independence of scales would then be examined by a factor analysis.

act in depriving a potential enemy of each resource. The data gathered from the second evaluation have not been analyzed because the analysis of the first evaluation yielded sufficient information for this pilot study.

The questionnaire was handed to each of the twenty-four subjects with a verbal request for its completion and return to the author.*

The responses to the questionnaires were tallied, and the means were calculated and plotted. See Figure III-1. Analysis of the resultant data revealed two interesting facts: (1) The world situations that were thought to have been arranged in order of severity were not so regarded by the subjects. "U.S. in Limited War," which we considered to be a rather drastic situation and which we ranked eighth in severity, was apparently considered much less severe by the subjects, who ranked it fourth with respect to resource allocation (concluded from inspection and manipulation of plots of responses). (2) Since the subjects' responses to a number of the thirteen value categories or resources were essentially similar, it was possible to group the thirteen into six categories. Figure III-1 shows the bunching of responses which led to consolidation into the following six categories:

Old Category	New Category
Goodwill Political Advantage	Political
Intelligence	Intelligence
Weapon R and D Industrial Capacity Technological Superiority	Technical/Industrial
Time Money Territory	Economic
Human Life	Human Life
Weapon Stockpile First Strike Capability Second Strike Capability	Military

*This is not the most efficient manner for conducting an experiment since it results in occasional long delays in the return of the questionnaires, as was the case here. However, it was the only possible method of operation with subjects who were located in different departments of the company and whose schedules and commitments varied widely.

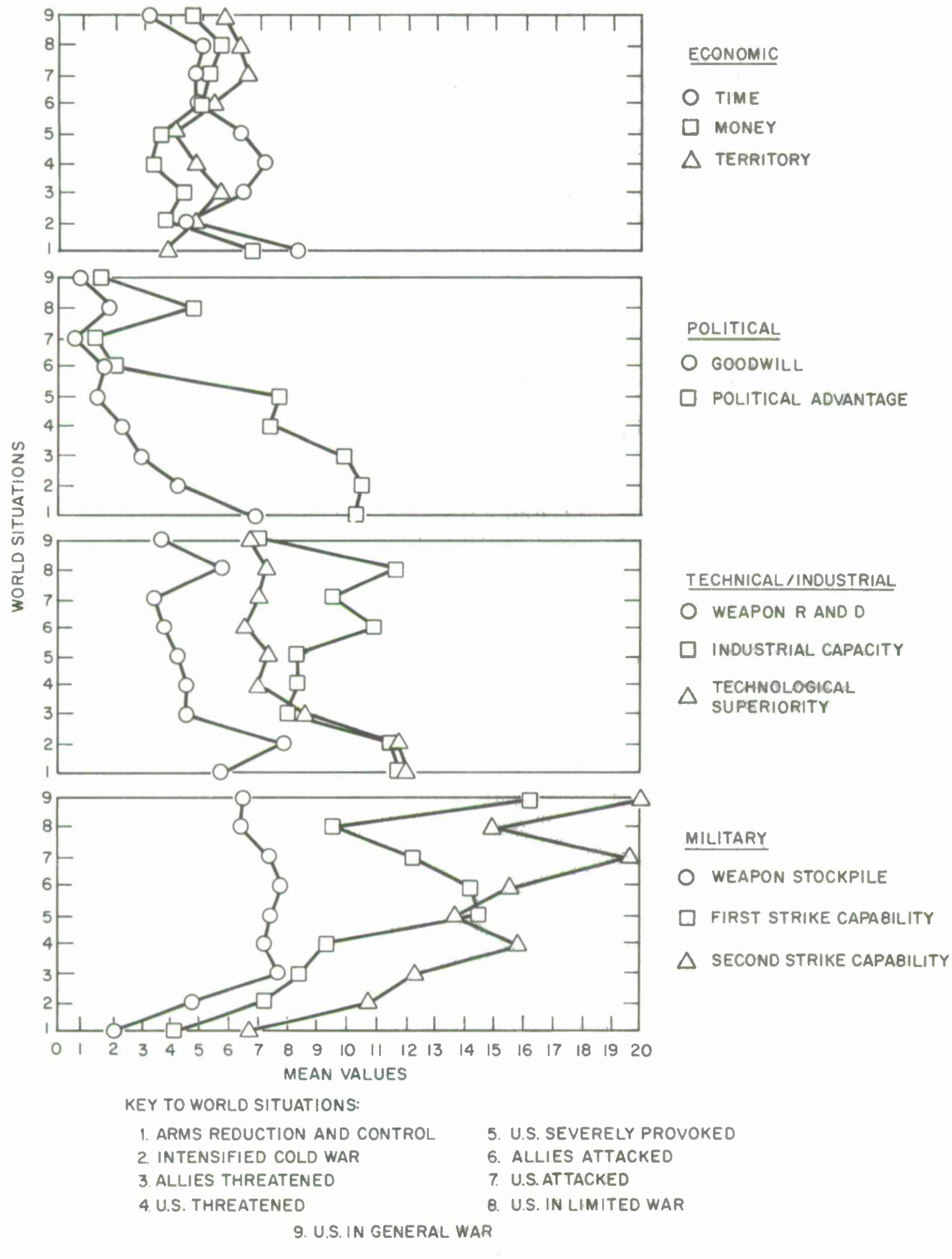


Figure III-1. Means of Subject Responses to Value Questionnaire

The means of the ults that the subjects assigned for these new combined categories for the nine world situations are shown in Table III-1. Figure III-2 is a graphical presentation of these means. The horizontal separation of the value lines shows rather clearly that the six categories appear to be distinct and different. This is particularly evident at the high end of the situation scale: that is, the more severe the world situation, the more distinct were the subjects' responses with respect to the value categories. The independence of five of the value categories for the six more important world situation was tested by means of the t-test for the significance of the difference of means of small samples. The results of the t-tests are shown in Table III-2. A probability of 0.05 was chosen as the cutoff probability for significance. That is, if the t-test indicated that there were five or less chances in 100 that the two samples were from the same population, we assumed that they were indeed different categories. It will be seen that for the least serious condition, "Arms Reduction and Control," only three of the ten pairs showed independence, whereas under the most severe condition, "U.S. in General War," seven pairs displayed independence. Further, Table III-2 shows that "Human Life" seemed highly

TABLE III-1

MEANS OF ULT VALUES ASSIGNED BY TEST SUBJECTS
(SHOWN FOR COMBINED CATEGORIES)

Value Categories	World Conditions								
	1	2	3	8	4	5	6	7	9
Political	8.40	7.16	6.34	3.30	4.59	4.48	1.72	0.88	1.16
Intelligence	14.01	11.48	13.85	12.37	13.93	13.21	12.81	10.13	9.24
Tech/Ind.	9.70	10.30	6.92	8.09	6.47	6.36	6.96	6.51	5.67
Economic	6.12	4.24	5.39	5.52	5.56	4.55	5.04	5.50	4.44
Life	8.60	7.76	8.48	9.16	10.00	9.40	9.96	13.20	14.44
Military	4.36	7.62	9.42	10.27	10.76	11.81	12.46	13.01	14.21

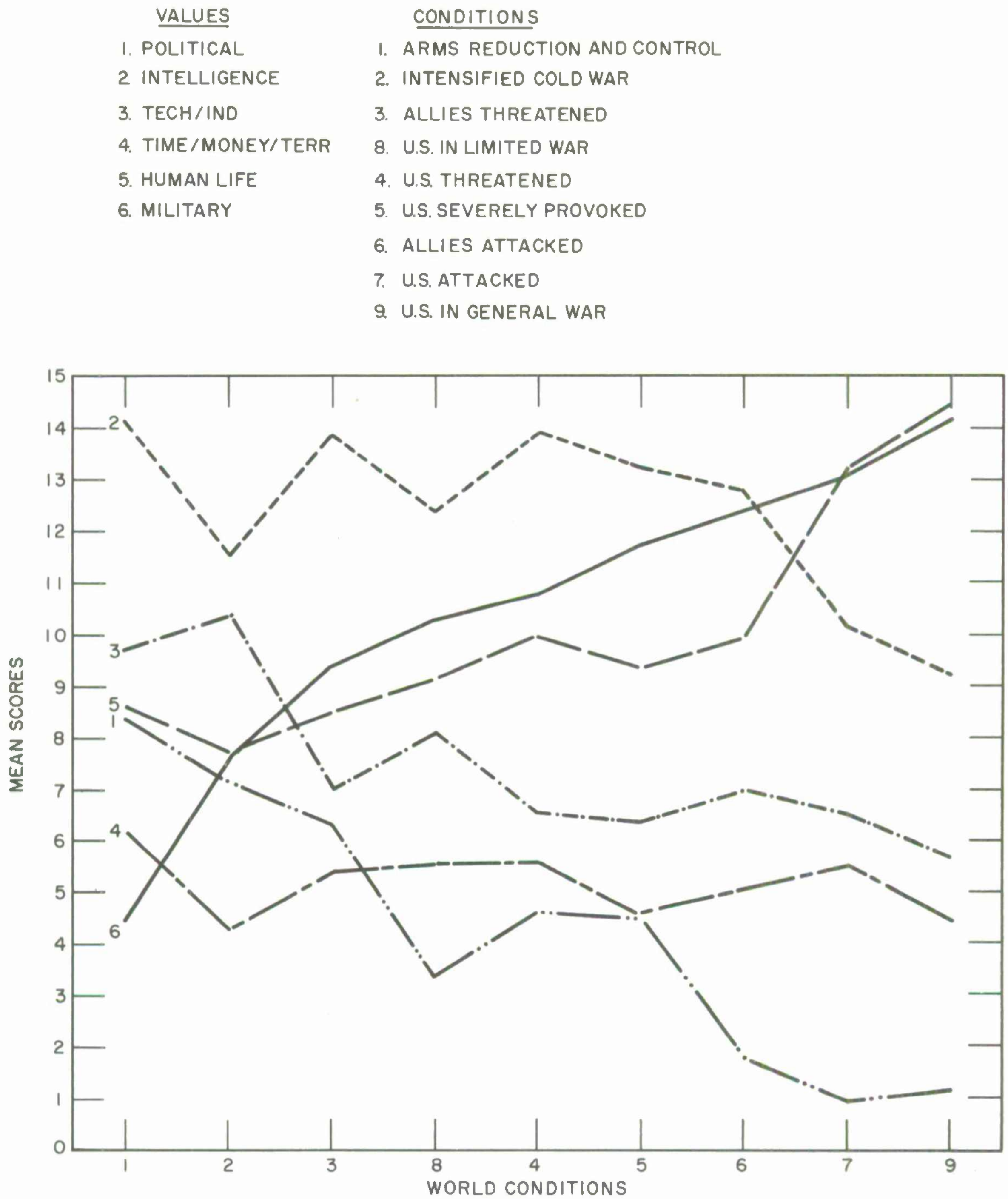


Figure III-2. Mean Scores of Ult Values for Six Combined Value Categories

TABLE III-2

INDEPENDENCE OF VALUE FACTORS t-TESTS*
(P = 0.05 is cutoff for significance)

Value	Situation	Human Life	Political	Tech/Ind.	Military
Economic	Arms Reduction & Control	N	N	.02	N
	Intensified Cold War	N	.001	.001	.01
	U.S. in Limited War	N	N	.05	.01
	Allies Attacked	.001	.01	N	.001
	U.S. Attacked	N	.001	N	.001
	U.S. in General War	.05	.001	N	.001
Human Life	Arms Reduction & Control		N	N	N
	Intensified Cold War		N	N	N
	U.S. in Limited War		N	N	N
	Allies Attacked		.02	N	N
	U.S. Attacked		.01	N	N
	U.S. in General War		.01	N	N
Political	Arms Reduction & Control			N	.01
	Intensified Cold War			.01	N
	U.S. in Limited War			.001	.001
	Allies Attacked			.001	.001
	U.S. Attacked			.001	.001
	U.S. in General War			.001	.001
Technical/Industrial	Arms Reduction & Control				.001
	Intensified Cold War				N
	U.S. in Limited War				N
	Allies Attacked				.01
	U.S. Attacked				.001
	U.S. in General War				.001

*Key: N = Not significant at .05 level
.01, .02, .05, etc. = significance level

related to the other categories and showed decided independence only from the "Political" category on the severe end of the situation scale. The "Military" and "Political" categories, on the other hand, displayed marked independence from each other and the other categories (except for the "Military/Human Life" pair, which was highly dependent).

In spite of its many shortcomings, the first questionnaire (which we shall refer to as the value questionnaire) produced some data that, upon analysis, turned out to be quite interesting. It showed that subjects' beliefs with respect to a class of intangibles, which we have called value categories, can be measured. However, in the time available we have not been able to show that they can be measured reliably. This test remains to be done. We have shown that subjects can handle these highly unstructured, loosely defined, and unquantified concepts and, in effect, rank them in order of importance (although there were two subjects for whom the test was too unstructured and who, therefore, refused to participate in the experiment).*

To measure our subjects' decision performance, so that we might correlate it with their value spectrum, it was necessary to construct a decision questionnaire. This questionnaire (pp. 94 to 100) contains ten military and quasi-military problem situations, each described in a separate paragraph and each followed by five alternate problem solutions. The questionnaire instructions asked the subjects to project themselves into each situation, in turn, to assume the role of a specific important decision maker, and then to rank the five alternatives in order of preference.

The same twenty-four subjects who had completed the value questionnaire were asked to fill out the decision questionnaire. The same form of questionnaire distribution and collection was employed as was used with the value questionnaire.

The subjects' responses were tallied and plotted and subjected to some basic statistical calculations. Inspection of the data immediately revealed that substantial agreement existed among the subjects with respect to the least preferred and most preferred alternatives, but there was less agreement on the three intermediate choices. In order to determine whether this trend toward agreement at the extremes and disagreement at the center of scale was statistically significant, we ran chi square tests on the data. Figure III-3 illustrates the results of this statistical

---

*We realize now that the questionnaire was too complex and included too many value categories and probably also too many world situations. Keeping track of the 100 ults as they were assigned proved cumbersome for the subjects. For any future questionnaire, we propose having the subject assign from zero to ten ults per category/situation. Before this type of questionnaire is used again, the six value categories and the set of world situations should be re-evaluated.

PREFER CATEGORY	LEVEL OF SIGNIFICANCE				
	.001	.01	.02	.05	NOT SIG.
1	7	2	0	0	1
2	1	5	1	1	2
3	2	1	1	2	4
4	2	2	1	1	4
5	9	0	1	0	0

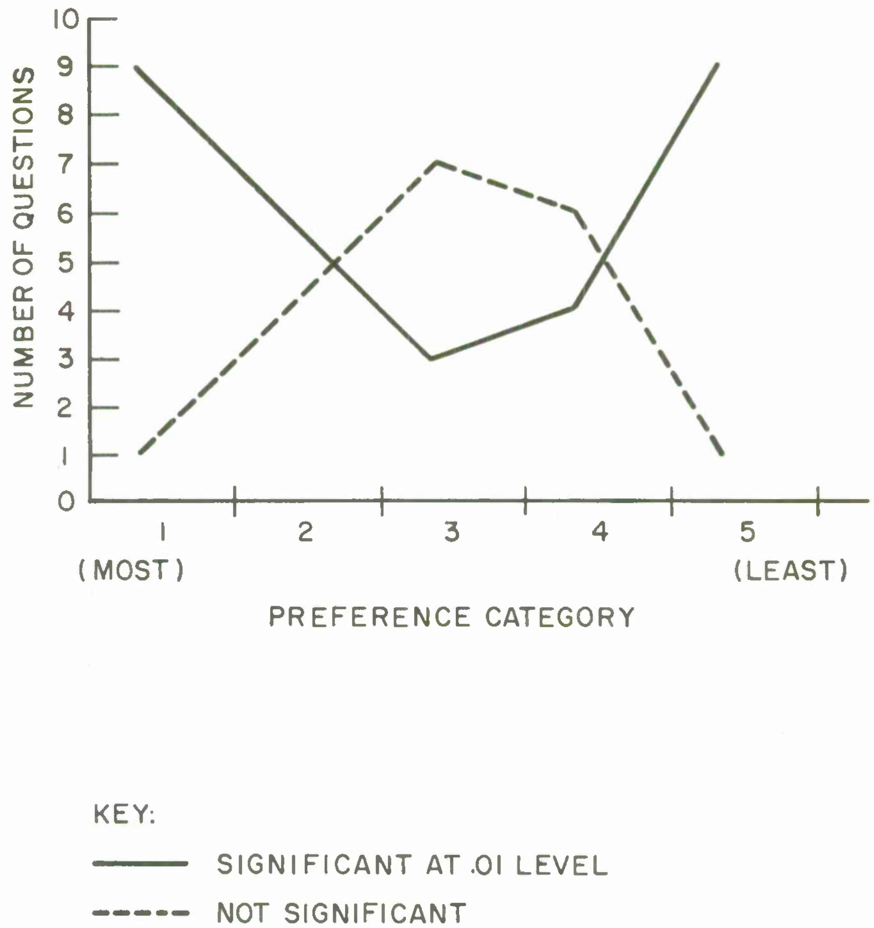


Figure III-3. Frequency of Significant Preferences Among Ranking of Alternatives of Decision Questionnaire analysis. It shows clearly a significant rejection of chance in nine of the ten questions of the most preferred item and in nine of the ten questions of the least preferred item. The middle choices show a much lower rate of rejection of the null hypothesis (no difference from chance expectance). We have, at present, no clue to the real reason for such marked agreement at the extremes of the preference scale. Two possible reasons are the following:

1. Homogeneity of the sample group. If the subjects, all drawn from an essentially confined environment, were sufficiently alike in the characteristics underlying their responses, this similarity could account for the agreement found.

2. Poor questionnaire construction leading to transparent problem situations with obvious solutions would cause the same kind of agreement at the extremes.

Data to substantiate either theory are insufficient.

To enable us to relate the subjects' value scales to their decision responses, it was necessary to determine the elements common to both questionnaires. This task could have been performed prior to the subjects' filling out the decision questionnaire, but we anticipated a delay in the return of these and decided to use this period of waiting in an effort to relate the two forms. This effort was not as valid as we would have liked because of the unavailability of independent expert judges. However, we did recruit as judges eight scientists at Technical Operations Research. These judges were asked to attribute one or more of four value categories to each of the five alternatives of each question of the decision questionnaire. The value categories were "Military," "Economic," "Political," and "Humanistic," roughly corresponding to four of the categories of the value questionnaire. The judges were instructed to assign to each alternative a value between -5 and +5 for each of the four categories. For example, the instructions for the "Military" value were as follows:

> Does the response have some military value with respect to the overall U.S. position and goals? This may be a negative value as well as a positive one, i.e., the response may be detrimental to the U.S. or it may be beneficial. For example, the deployment of a fleet into a war zone may be advantageous if it results in, say, the safe landing of troops on an objective island. However, if the deployment should result in the sinking of a cruiser (U.S.), it would be detrimental to the U.S. effort.

The instructions for the other value categories were similar. The judges' responses were tabulated and the means were calculated. The means are presented in Table III-3. Somewhat arbitrarily, we decided that a mean in excess of ±1.875 indicated a "significant" presence of a value in an alternative, providing that there was considerable agreement among the judges.*

TABLE III-3

MEAN VALUES ATTRIBUTED TO DECISION ALTERNATIVES BY JUDGES

Question	Decision Value Factor				Question	Decision Value Factor			
	Milit.	Econ.	Pol.	Hum.		Milit.	Econ.	Pol.	Hum.
1. a	-3.125	-1.125	-4.0	-0.625	6. a	+2.175	+0.75	+0.125	+2.375
b	+2.25	0	+0.125	+1.375	b	+3.875	-0.625	+0.25	-1.0
c	-1.25	0	+1.625	-2.0	c	+3.5	-1.25	-2.5	-3.0
d	+4.5	+0.5	-1.125	-0.375	d	+1.125	+0.25	+2.75	+2.125
e	-0.5	+0.125	+3.0	+1.875	e	-2.0	-0.375	-0.25	+0.375
2. a	+2.25	+1.875	+1.625	-3.375	7. a	+1.875	+1.875	+2.625	+0.875
b	+3.25	+0.875	+1.125	0	b	+2.125	+0.875	+1.375	+0.5
c	+0.875	+1.0	+0.25	-2.375	c	+0.25	-1.25	-1.25	-1.0
d	+2.75	+1.75	+1.125	-0.5	d	+2.25	+0.625	+0.375	-0.625
e	+4.125	+0.25	+2.125	+1.5	e	+0.125	+1.375	-2.875	-1.625
3. a	-1.5	-2.5	-1.25	-0.875	8. a	-3.5	-0.25	-2.375	-0.125
b	+1.875	+1.25	+0.625	+0.625	b	-3.25	-0.125	-3.5	-0.625
c	+1.875	+1.0	+0.5	+0.375	c	+0.375	+0.625	-0.5	+1.25
d	+3.25	+0.875	+1.0	+1.125	d	+0.75	+0.5	+1.0	+1.0
e	+0.5	+3.0	+3.0	+3.25	e	+0.375	+0.375	-0.25	+1.5
4. a	+0.875	+0.25	+2.0	+3.625	9. a	0	-0.125	-0.5	-0.5
b	0	-0.125	-0.5	-1.75	b	-0.75	-1.0	-2.125	-0.625
c	+0.125	+0.25	-1.0	+0.5	c	-0.5	-0.25	-1.875	-0.25
d	+0.25	0	0	-0.125	d	-0.25	0	-1.875	-1.125
e	+0.125	0	0	+0.375	e	+0.125	0	+1.25	+0.125
5. a	+1.375	+0.875	+1.5	+1.25	10. a	+2.125	-1.125	+1.375	-1.0
b	+2.5	+0.5	0	0	b	-0.125	-0.625	+1.25	+0.125
c	+2.375	+2.25	+0.125	-0.125	c	+0.625	-1.125	-0.75	-0.5
d	+1.375	+0.25	+0.75	-0.375	d	-0.25	-1.5	-0.875	-0.5
e	+2.0	+0.375	+0.375	+0.25	e	+1.5	+0.25	+0.125	-0.5

*The mean of 1.875 for "Military" value in item 7a, for instance, was not deemed significant because three of the eight judges attributed 0 "Military" value and two a value of only +1.

Using the judges' attributed values ("significant means"), we next calculated each subject's individual value indexes as reflected by his responses to the decision questionnaire. This was done by assigning for each first and second ranking (of alternatives) in the decision questionnaire the "significant" means obtained from the judges (for those alternatives for which there was a "significant" mean). We used both the first and second ranking because we believed that the addition of the second ranking would introduce more individual differences, since so much agreement had been displayed for the first ranking. The values obtained for each question for each subject were then added to obtain four total value scores for each person ("Military," "Economic," "Political," and "Humanistic").

We then had for each subject two sets of comparable value scores we could correlate to determine the relationship between the value and decision responses. Since we did not have time to correlate the data for all nine world situations used in the value questionnaire, we attempted to determine the subjects' views of the world situation today in the belief that their responses with respect to the situation today would be the most valid. Accordingly, we requested twenty of the original twenty-four subjects to indicate, on a short questionnaire, which of the nine world situations represented most closely that in which the U.S. finds itself today. The response was as follows:

World Situation	N	%
Arms Reduction and Control	5	25%
U.S. in Intensified Cold War	6	30%
Allies Threatened	1	5%
U.S. in Limited War	7	35%
U.S. Severely Provoked	1	5%
	20	100%

As a result of the subjects' responses to the questionnaire on world situation, we chose "Arms Reduction and Control," "U.S. in Intensified Cold War," "U.S. in Limited War," and as a control, "U.S. in General War," for our correlation conditions. The Pearson product moment correlation was utilized and the following pairs were correlated:

Value Factor	Decision Factor
Military	Military
Technical/Industrial	Economic
Economic	Economic
Political	Political
Human Life	Humanistic

Table III-4 lists the correlation coefficients for these factors under the four world situations (value questionnaire). Only factors that appeared to be related intuitively were correlated. As the data show, however, none of the correlation coefficients is particularly high, although a number of them do indicate an encouraging trend. Considering the pilot and pioneering nature of this experiment, which is really a first trial of a trial-and-error approach, these coefficients are not too discouraging. It will be seen also that, in general, the coefficients for the three "prevailing" world situations are higher than those for the control situation, but still not statistically significant.

What, then, is the reason for these low correlation coefficients? We cannot be sure at this time, but a number of explanations are suggested:

1. There really is no marked relationship between a person's value spectrum and the way he makes command and control type decisions.

2. The sample of subjects used in the experiment is atypical of the subjects with whom we should be concerned — the military decision makers.

3. The value and/or decision items of the questionnaires were poorly chosen and do not reflect those factors that really possess a predictive relationship.

4. The small sample size of the experiment unduly influences the statistics because of the inordinately large effect of a few uncharacteristic responses.

5. The value factors attributed by the non-expert judges to the decision alternatives were not properly assigned.

TABLE III-4

PEARSON PRODUCT MOMENT CORRELATION COEFFICIENTS

Value Factors	World Situation	Decision Factors			
		Military	Economic	Political	Humanistic
Military	Arms Reduction & Control	.25			
	Intensified Cold War	.33			
	U.S. in Limited War	.32			
	U.S. in General War	.28			
Technical/Industrial	Arms Reduction & Control		-.30		
	Intensified Cold War		.03		
	U.S. in Limited War		-.10		
	U.S. in General War		-.08		
Economic	Arms Reduction & Control		.01		
	Intensified Cold War		.14		
	U.S. in Limited War		.33		
	U.S. in General War		.30		
Political	Arms Reduction & Control			.40	
	Intensified Cold War			.51	
	U.S. in Limited War			.26	
	U.S. in General War			.23	
Human Life	Arms Reduction & Control				.18
	Intensified Cold War				.25
	U.S. in Limited War				.24
	U.S. in General War				-.08

Correlation would be statistically significant at:
the .05 level if the coefficient were $\geq$ .343*
the .01 level if the coefficient were $\geq$ .485

*Sidney Siegel, "Nonparametric Statistics for the Behavioral Sciences" (New York, N.Y.: McGraw-Hill, 1956).

SUGGESTIONS FOR FUTURE WORK

A great deal of further research is required to determine which of the above or other explanations is responsible for the low correlation coefficients. We like to believe that the first explanation is untrue. The general, though low, relationship found does indicate that it may be untrue. Further experimentation is needed with larger and more appropriate (i.e., military) subject groups, expert judges, and completely reworked questionnaires.

If this research were to be pursued until the role of value judgment in command and control system decision processes were fully understood and the pertinent value factors were isolated, the resultant findings would have considerable implication for command and control exercise and evaluation technology. We believe that it would then be possible, through training and exercising, to instill the desired and pertinent values in the responsible commanders, thus assuring a reasonable uniformity in the value component of the decision process. Periodic evaluation of this uniformity could be achieved with measurement techniques that would evolve from the research.

VALUE QUESTIONNAIRE

You are being asked to take part in an experiment. Imagine that you have been appointed Chairman of the Joint Chiefs of Staff of the U. S. and have received a directive from the President to indicate to him how the country's military resources and potential should be invested so that the nation may be prepared for any eventuality.

You are given a stake of 100 ults* per situation which you are to distribute among the resources available for each world situation. Remember that you are assigning <u>value</u>, real, intangible, etc., not money.

There are two tables attached on which you are to indicate your value assignments. On each you will find the World Situations listed along the left side. Be sure to consider each situation separately and independently. Across the top of the page are listed the Value Factors in which you are to invest your 100 ults.

You are being asked to invest in two "portfolios." The first (Table 1) might be considered as an insurance portfolio. That is, you should invest your 100 ults per situation with the point of view in mind: "How much should I invest in protecting each against loss?" In the second table consider how much each resource is worth (in each world situation) if you could deprive the enemy of it. For example, how much would it be worth if you could take enemy territory away from him in a limited war situation.

Now, go ahead. Remember to consider each situation separately, and invest your full 100 ults for each situation.

---

*A unit of value which does <u>not</u> correspond to money.

VALUE OF U. S. RESOURCES

Resource / World Situation	Time	Money	Human Life	Territory	Goodwill	Political Advantage	Intelligence Data	Weapon Stockpile	Weapon R and D	Industrial Capacity	Technological Superiority	First Strike Capability	Second Strike Capability
Arms Reduction and Control													
Intensified Cold War													
Allies Threatened													
U. S. Threatened													
U. S. Severely Provoked													
Allies Attacked													
U. S. Attacked													
U. S. in Limited War													
U. S. in General War													

QUESTIONNAIRE TABLE 1

VALUE OF ENEMY RESOURCES

Resource / World Situation	Time	Money	Human Life	Territory	Goodwill	Political Advantage	Intelligence Data	Weapon Stockpile	Weapon R and D	Industrial Capacity	Technological Superiority	First Strike Capability	Second Strike Capability
Arms Reduction and Control													
Intensified Cold War													
Allies Threatened													
U. S. Threatened													
U. S. Severely Provoked													
Allies Attacked													
U. S. Attacked													
U. S. in Limited War													
U. S. in General War													

QUESTIONNAIRE TABLE 2

DECISION QUESTIONNAIRE

You are being asked to take part in the second phase of an experiment on decision making in Command and Control Systems.

On the following pages you will find ten separate descriptions of hypothetical political or military situations. You are asked to assume the role of an important decision maker who must choose the most appropriate solution to each situation, ranking the five alternatives in order of preference. Assign a 1 in the box to the left of the preferred solution, a 2 to your second choice, and so on, with 5 being assigned to the solution preferred least.

1. You are the commander of the U. S. Forces in Germany. You are informed that the enemy is planning to attack West Germany with 25 divisions and that this attack will be preceded by heavy bombardment of your front line troops with gas filled shells. Your men have no gas masks. Would you:

- [ ] a. Evacuate your troops and yield the country without a fight.
- [ ] b. Attempt to procure masks as quickly as possible and at any cost.
- [ ] c. Order your men to hold the line no matter what happens.
- [ ] d. Strike the enemy before he can attack you.
- [ ] e. Relay warning of impending attack to the President and suggest indicting enemy in U. N. in hope of warding off attack.

2. You are the Strategic Air Command representative to the Joint Navy - Air Force Targeting Board. You have been asked to recommend a new target for a newly emplaced ICBM. Which one of the following five would you choose?

- [ ] a. Moscow — population 6 million.
- [ ] b. Vladivostok — 20 submarines, 3 cruisers, 12 destroyers, and several docks and shops.
- [ ] c. Leningrad — several Institutes of the Academy of Sciences.
- [ ] d. Magnitogorsk — 3 large steel mills.
- [ ] e. ICBM launcher complex on Kola Peninsula.

3. You are the chief of a unified (Army-Navy-Air Force) weapon research laboratory. A large project has just been completed, and you have twelve senior scientists ready for assignment to a new project. Would you:

- ☐ a. Lay off the scientists.
- ☐ b. Employ them on development of a light bullet-proof vest for the field soldier.
- ☐ c. Have them develop a miniaturized transmitter for U. S. Intelligence agents operating behind enemy lines.
- ☐ d. Add them to a project devoted to the development of decoys for the Minuteman missile.
- ☐ e. Loan them to the University of California to help develop an economical desalinization plant for underdeveloped countries with water shortages (and access to an ocean).

4. You are the engineer on an Apollo space craft orbiting the moon while the pilot and copilot have descended to the moon surface in the Lunar Exploration Module (LEM). They are due to return to the Apollo craft but have received a valve malfunction indication in their engine start procedure. You are rapidly approaching your earth return point, and if you were to make one more orbit around the moon waiting for the LEM to return, you stand only a 50-50 chance of having enough fuel to return all three men to earth. Would you:

- ☐ a. Risk one more orbit to give the pilot and copilot a chance to repair the valve.
- ☐ b. Return to earth as planned.
- ☐ c. Radio trouble to Apollo Control Center in Houston and request instructions, thus exposing failure to the world.
- ☐ d. Advise crew to attempt engine start risking that valve failure indication was spurious—if it is real, explosion may result.
- ☐ e. Ask pilot on the moon for instructions.

5. You are the military advisor to the local Vietnamese Army commander. One company of regular army troops has just been assigned to your command. How would you advise the commander to deploy these new troops in the face of indications that the Viet Cong are increasing their raids in your province?

- ☐ a. Deploy the company around the largest town in the province.
- ☐ b. Deploy the company to guard the local weapon arsenal.
- ☐ c. Deploy the company to guard the only munitions factory in the area.
- ☐ d. Disperse the company among the villages, dress them in civilian clothes, and have them gather intelligence of Viet Cong movements.
- ☐ e. Break the company up into platoons and assign them to a, b, c, and d as best as possible.

6. You are the Chairman of the Joint Chiefs of Staff. You have just been told that all enemy ICBM's have been placed on their launch pads and are being fueled. No bomber activity has been observed. What action would you recommend to the President?

- ☐ a. Alert the civilian population to seek shelter.
- ☐ b. Send the SAC airborne alert aircraft on their way to their targets and get the remainder of SAC airborne.
- ☐ c. Launch our own ICBM's and Polaris missiles.
- ☐ d. Ask the enemy to remove the missiles from the pads or you will do a, b, and c.
- ☐ e. Wait for further developments.

7. You are a high ranking officer at Hq. USAF charged with the responsibility of disposing of 30 obsolete and aging B-47's. You have bids from five countries to purchase the aircraft. To which one would you sell them?

☐ a. Australia—Could support the aircraft in their old age and would probably buy some more modern aircraft from the U. S. at a later date.

☐ b. Venezuela—Could probably support the aircraft. We would like to bolster the government. The acquisition of the aircraft would make the Venezuelan Air Force the most powerful in South America.

☐ c. Formosa—It would probably cost the U. S. aid money to maintain the aircraft, and it is not at all clear that the Formosan Air Force needs longer range bombers—might make Chiang more aggressive than we would like him to be.

☐ d. Spain—Although we do not really approve of Franco's dictatorial regime, we are offered an extension of our base rights treaty in exchange for the bombers.

☐ e. Union of South Africa—Would pay cash for the aircraft and could presumably support them, but the sale of the B-47's to the Union of South Africa would make the U. S. very unpopular in the other African nations, particularly those south of the Sahara. The Union of South Africa has, however, offered the highest price for the aircraft.

8. You are a senior DoD staff member. You have been told by Secretary McNamara that Russia has offered to completely withdraw its forces from Cuba in exchange for one of five U. S. concessions. You are to recommend to the Defense Secretary which one of the concessions should be conveyed to the President as being the most acceptable to the DoD.

- ☐ a. Cease Cuban overflights.
- ☐ b. Evacuate Guantanamo.
- ☐ c. Reinstate the Cuban sugar quota.
- ☐ d. Re-establish diplomatic relations with Cuba.
- ☐ e. Lift the Cuban trade embargo.

9. You are a high level White House Foreign Affairs advisor. The Russians have arrested a group of U. S. businessmen on tour of Russian factories on trumped up spy charges. You are asked by the President which one of the following five actions we should take.

- ☐ a. Negate our cultural exchange program with the U.S.S.R.
- ☐ b. Cease all trade with the U.S.S.R.
- ☐ c. Threaten diplomatic break with the U.S.S.R. unless the group is set free immediately.
- ☐ d. Arrest a Russian group of musicians touring this country.
- ☐ e. Confine our action to a formal protest through the normal diplomatic channels.

10. You are the Secretary of State of the United States. Indonesia has launched a large scale attack against Malaysia. Because our allies, Great Britain, Australia, and New Zealand, have gone to the aid of Malaysia and also because much of our population abhor the attack, we feel obligated to take some action to help the Republic of Malaysia. Which U. S. action would you recommend to the President?

- ☐ a. Actively support Malaysia by sending the Seventh Fleet and its Marines into combat on its side.
- ☐ b. Break diplomatic relations with Indonesia.
- ☐ c. Blockade Indonesia—this is a difficult task due to the large number of islands and the long distance involved.
- ☐ d. Cease all trade with Indonesia, which would damage the U. S. also, since we depend on Indonesian tin, bauxite, nickel, and manganese.
- ☐ e. Finance a plot to assassinate Sukarno.

APPENDIX IV

SEMANTIC MODEL*

INTRODUCTION

Many aspects of command and control system activities appear to involve the ability to understand meanings. We originally began to investigate the notion of meaning in an attempt to measure the difficulty of exercise scenarios. We felt that such a measure was important if one was to measure the calibre of a system's performance on the basis of exercise results; a system performing well on a simple exercise was not necessarily better than one not doing as well on a more difficult exercise.

Since we were dealing with the 473L system, which receives most of its messages in something that approximates ordinary English, we found that the way in which information was stated in incoming messages had considerable influence on how easily the information could be understood and assimilated into the system's image of the world (Appendix I). Thus, if a piece of information that was later to become important for problem solution were sent in a message that contained a great deal of other information, it was less easily assimilated in the memory of the people who read it than if it were sent in a short message containing little else.† Also, information that appeared after the problem had been stated seemed to be relevant, and was more likely to be recalled than information that appeared before the statement of the problem.

Since message features that were related to meaning seemed to be important parameters in determining the degree of difficulty of an exercise, we attempted to devise some measure of the clearness of the meaning of a message. This was not

---

*Part of this appendix was presented at the 2nd Annual Meeting of the Association for Machine Translation and Computational Linguistics in Bloomington, Indiana, on 30 July 1964.

†There were situations where this was reversed. A message that contains only one piece of information not relevant at the time of receipt may be completely ignored. The message that contains many pieces of information may be read more carefully, and even the parts that appear to be irrelevant at the moment may be stored for later use in the memory of the people who read it.

easy. It appeared that the clearness of a message was a feature, not only of the message itself, but also of (1) the representation of the state of the world within the system and (2) the way in which people individually, and the system as a whole, manipulated this representation.

For the purposes of this appendix, we will call the framework underlying such a representation a semantic space. Such a space is the matrix in which people and command and control systems represent their images of the external world. At some given time t, this space will contain a certain configuration of points C_t. If the system that contains this configuration receives some message M, the state of this space can be changed by that message into a configuration $U(M, C_t)$, which depends both on the message and the original conditions of this space. This change in the configuration represents the system's understanding of the message.

In this appendix we describe an initial classification of the overall structure of a semantic space within which such a representation might be made. This taxonomy was based on an analysis of how people handle meanings.

We are concerned with how one relates information in one part of a system's inputs to information from another part. If a system receives a message that the international situation forbids us to fly over some country X, how does it call up this fact when it notes later that some plan Y, which calls precisely for such an overflight, is to be activated? It is easy to say that the system has the message stored at the right place or that the human beings within it have some sort of magic machinery for retrieving facts according to relevance, but these statements do not answer the question. Fact X has to be stored in such a way that it will be found when the system looks into the storage system under the category: plan Y. If one considers his occasional difficulties in using the telephone book,* he can imagine the difficulties that a more general filing scheme might encounter. In spite of these difficulties, a person seems to require extraordinarily little time to recall an

*For example, if a man wants to enlist in the Air Force, he cannot find the telephone number to call under A (for Air Force or Armed Services), nor under E (for enlistment), nor under R (for recruiting station). It is listed under U (for United States Government).

incredible variety of facts under the most varied of circumstances.* It is the basis of this ability that we seek to study.

If a suitable representation can be found it can provide a basis for one part of the deductive inference model of Appendix I; it can provide a basis for procedures for sound inductive inference using the model of Appendix II; and it can provide a sounder basis for studying the way values are brought to bear on decisions as discussed in Appendix III.

ELEMENTS OF A SEMANTIC THEORY

We shall attempt to describe the semantic machinery of systems, and of people within systems, as machinery that manipulates strings in terms of form alone.

A semantic theory can be looked at as a black box (Figure IV-1) that determines (and may compute) some semantic function of a discourse† in a natural language or

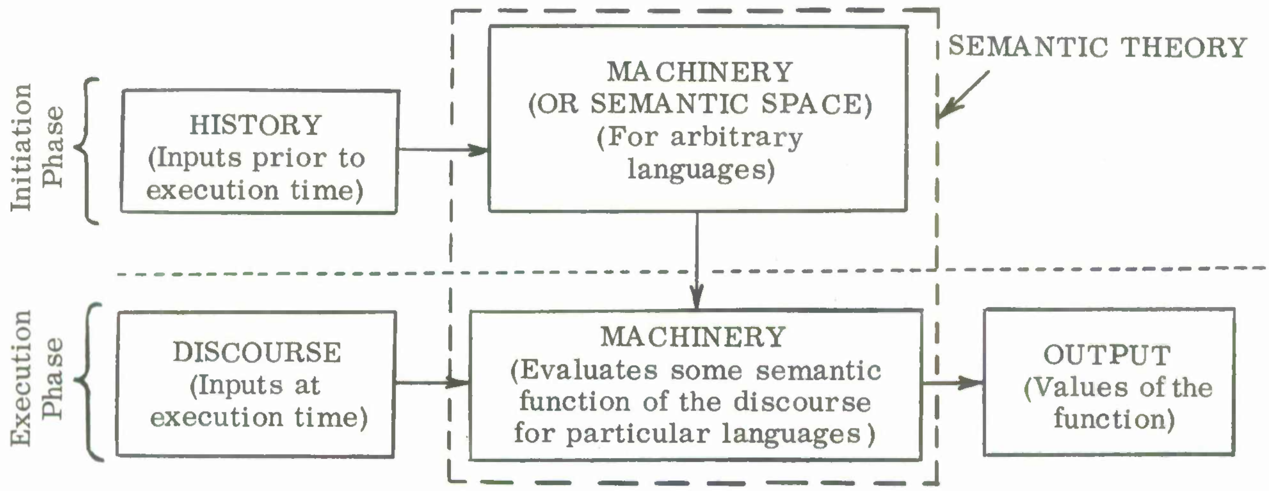


Figure IV-1. Features of a Semantic Theory

*Some facts appear to become blocked from recall even when they should be recognized. This is probably due to unconscious factors, which are a subject for psychiatrists and are not dealt with here.

†A discourse is defined as a finite set of utterances or strings.

languages. A larger body of facts (the "history"[*] of the device) in some manner orients the theory to some particular language or languages. Associated with these two types of input are two types of machinery, which are not necessarily distinct. The first machinery transforms the history into something that the second machinery uses in determining the required semantic function. These two phases of operation, which may overlap, are further defined as follows:

> The <u>initiation phase</u> sets the parameters of the machinery for a particular language or group of languages.
>
> The <u>execution phase</u> determines a particular value of a semantic function for a given discourse.

A semantic function may thus be looked at as a function of two arguments: (1) a relatively small discourse and (2) the "linguistic competence" developed from a given history. One example of such a semantic function is a human interpreter. Assume that such a person has been trained in two languages, L and M. This training is the result of his having dealt with an extensive history, only parts of which consisted of fragments from L and M. The rest of the history came to the interpreter in terms of his other (nonlinguistic) experiences. (Human beings do not speak a natural language until enough nonlinguistic experience has been built up to give them something to talk about.) The interpreter is capable of taking utterances of L and transforming them into utterances of M that have similar meanings. The concrete results of this process are the evaluation of a semantic function of given utterances of L with values in M. One way of looking at such a performance is to consider languages as the class of all their allowable utterances. Given some utterance selected from L (e.g., "moi"), the translation function may be looked at as a selection function over M. The utterance "to me" is selected as a value. A more sophisticated function would select the class of all possible utterances that might have similar meanings in M (e.g., "me," "myself").

---

[*]The "history" may include linguistic and nonlinguistic facts. It may also include discourse about the discourse.

SOME TYPES OF SEMANTIC THEORIES

The initiation phase of a semantic theory communicates to the application phase by producing some sort of a change in the latter's machinery or semantic space. One example of a semantic theory in which this semantic space is made explicit is the theory that underlies the use of a word-for-word dictionary to translate between two languages L and M. Such a dictionary is the link between the initiation (performed by one person) and the execution (usually performed by another person). To simplify the example, assume that the dictionary contains only words and no phrases of any sort. Such a dictionary is a simple mapping from words of L to words of M. To remove the grammatical component,* let us assume that the inputs to the dictionary user are the parsed strings of a discourse in L, and that the outputs are some sort of kernels in M, with appropriate grammatical markers attached to provide the required inputs for generating a grammar of M. The dictionary now is a part of a semantic theory with the following features:

> Inputs from the Discourse: The inputs from the discourse are taken in one word at a time. Contextual reference enters by means of grammatical markers.
>
> History: The history required, which is the input to the linguist who compiled the dictionary, is indefinite. If the linguist is lucky, he is bilingual or has available some sort of Rosetta Stone. In both cases, however, the supposed starting point obscures the fact that the information required is both linguistic (i.e., utterances in L and M) and nonlinguistic. This fact is clear if the dictionary has to be compiled by investigating the conditions under which native speakers of M make certain utterances, and then comparing these with the conditions under which native speakers of L make other utterances. Initially at least, these conditions are not linguistic, or at least they are not in L or M. When the linguist is bilingual and no native speakers are available, the augment still holds that the translation requires inputs that are not strictly in L and M.

---

*Usually covered by a page or two at the beginning of the dictionary.

Initiation: The device used to store the history for "programming" the machinery that is to do the actual translating is a dictionary. Such a dictionary is a simple piece of machinery. It can be described as a context-free, nondeterministic grammar, all of whose transformations are terminal.

Machinery: The machinery for implementing such a dictionary may be either (1) a machine capble of executing the mapping defined by the dictionary or (2) a person who uses the results of such a mapping, together with his linguistic intuition (initiated separately), to produce a translation. In general, the latter produces a more satisfactory translation because a person can select, from the several possible target discourses, the one that makes sense to him; or he may use his intuition to make a guess not even suggested by the literal application of the dictionary.

Varying the values of these four parameters can produce descriptions of different semantic theories at approximately the same level of generality. The nature of the variations gives some measure of the strength or weakness of these theories in various directions, and permits comparisons between them. We shall discuss these variations briefly, using as an example a simplification of a class of semantic theories that occurs frequently in information retrieval. In this class of theories, the thing passed between initiation and execution is some function of the frequency of the co-occurrence of words within the sentences of some given history.

VARIATION OF THE INPUTS FROM THE DISCOURSE

In the theory of the dictionary, the inputs from the discourse are parsed strings, examined word-by-word. In the simplified information retrieval theory, the inputs are sentences, examined as sets of words. These theories iteratively define some semantic function of their inputs, and each iteration makes use only of a limited amount of information about the discourse. These theories can be compared with respect to (1) what kinds of information they use and (2) how much of it they use.

In this example, the theories are identical with respect to the distance at which contextual information about the discourse can be relevant: in both cases the limits are to the ends of the sentence.* The two types of theory differ with respect to the types of information that are relevant beyond the individual word being considered in an iteration. In the dictionary-based theory, the only information that is relevant beyond the structure of the given word might be called syntactic or grammatical information. (For example, in translating the single word pendule from French into English, one might like to make use of the gender determined by the preceding article in the discourse.) The information used by the sample information retrieval theory, on the other hand, might be called semantic. (Thus, one uses the specific forms of the other words in the sentence; e.g., they are spelled t-a-b-l-e or F-o-u-c-a-u-l-t.) See Figure IV-2.

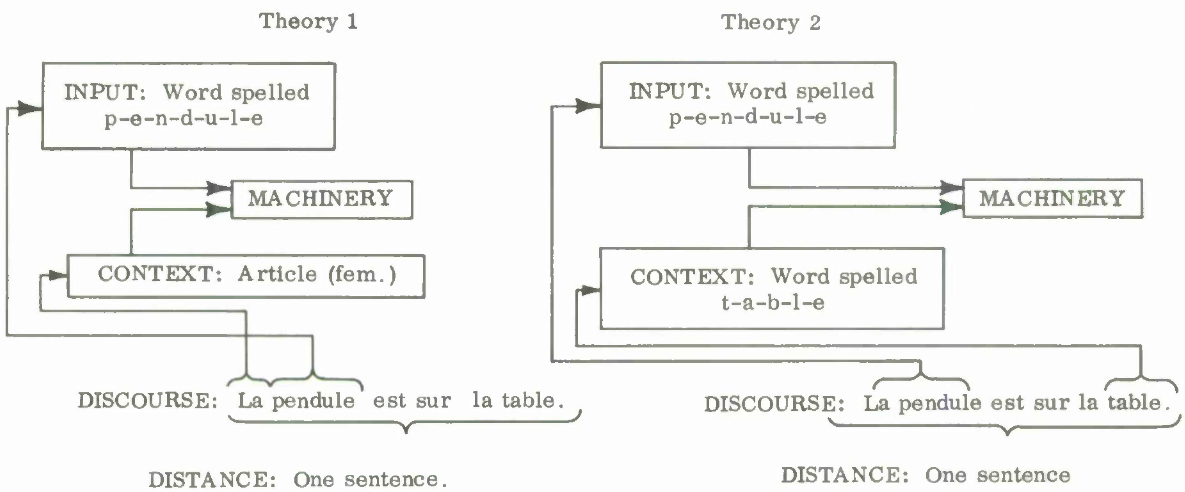


Figure IV-2. Distance and Type of Input for Two Semantic Theories

*This is not true of all similar theories used in information retrieval or in machine translation. However, this "distance" is a variable in terms of which such theories can be described. Later we shall suggest that this distance ought to be arbitrarily great. Ideally, such a distance should be sufficient to allow a translator with the unlikely task of translating "The Brothers Karamazov" into pre-Dostoevskian Hungarian to use the fact that Alyosha was Dimitri's younger brother, mentioned early in the book, to translate the Russian word for "brother," near the end of the book, into the appropriate Hungarian word. He would need this information to resolve the choice between using the word "elder brother" or the word for "younger brother," since pre-Dostoevskian Hungarian lacked a single word for "brother." This distance should also be sufficient to account for how an operator in a command and control system uses his training (which may have occurred years before he uses it) in his task.

The sample semantic theory used in translation treats words one at a time (e.g., it translates "pendule" into "clock"). In this treatment it can use the form of the word (it is p followed by e ...) and the grammatical context within the sentence (it is preceded by "le" and therefore a masculine noun). The sample information retrieval theory might treat an entire sentence (e.g., why does tantalum corrode?)* as sets of words perhaps in alphabetic order: corrode, does, tantalum, why). If one uses the word "semantic" to describe information that deals with the specific forms of words, one might say that this information retrieval theory uses sentence-wide semantic context and no grammatical information.

VARIATIONS OF THE INPUTS FROM THE HISTORY

In the general class of information retrieval theories we are using as an example, the inputs from the history are the same as the inputs from the discourse; namely, co-occurrence of words in sentences. In general, a statistical function of the frequency of such co-occurrence is computed and this, in turn, provides the inputs for the execution. The two phases use the same sort of machinery, but they use little information from the history, much less than would be used by a human being.

There appear to be languages in which our sensory organs communicate to the central storage unit within our heads that are somehow similar to the spoken or written languages with which linguists deal. One of the big problems of semantic theories is to account for the ways in which these "internal" languages are related to the spoken or written languages.

VARIATIONS OF THE MACHINERY USED IN INITIATION

The theory that utilizes the word-for-word dictionary requires a human being for initiation (making the dictionary). The general class of information retrieval theories we have been using as an alternative example uses a computer (or some other sort of mechanical or electrical device) for computing the statistical functions of the history that constitutes the outputs from initiation. Although the range of

---

*Jessica Melton, of Western Reserve, has pointed out that this example is ambiguous. It is not clear whether one wants to know why tantalum corrodes other things or why it is itself corroded.

material such theories use for the inputs of initiation is narrow (and the theory is therefore weak), the explicitness of the machinery might well be considered a point of strength. One could argue that the less given in the way of machinery used both for initiation and for execution, the stronger the theory, at least in this particular direction.

VARIATIONS OF THE MACHINERY USED IN EXECUTION

The effect of using more powerful (or initially more complex) machinery for execution is demonstrated in the dictionary. If we use machinery that simply replaces the words in the discourse according to the given dictionary, the results may seem smooth if enough attention is paid to the grammatical form of the results, but the sense of the original is seldom retained. If we use human machinery, with its more complex history, we can get a better result, since the human being can use information in ways we cannot yet make explicit.

FEATURES OF MEANINGS AND THEIR MANIPULATIONS

In the preceding section we sketched some considerations bearing on the more abstract features of machinery used in some semantic theories. In this section we shall consider some of the abstract features of the meanings with which this theory deals.

The question "what is meaning?" makes most people shudder. However, this question seems to be no worse (and no better) than such questions as: "what is force?" and "what is motion?" and it can be dealt with in much the same way. When a physicist deals with the latter two questions, he does not go into a philosophical trance to divine the nature of these things. Instead he abstracts from the properties of these things, tries to describe them, and then tries to find some abstract mathematical object that has similar features. From then on, he deals with the mathematical object instead. Suppose one tries to do the same kind of thing with meanings. The first task is to discover some of the features of meanings.

Consider the following sentence:

(1) This is a bow.

We shall be concerned with things that can be done to this sentence that have some effect on the meaning of the word "bow." In other words, we want to consider what sorts of things change meanings without really considering what is being changed.

One of the features of the word bow in (1) is that it has more than one meaning. Consider two meanings of the word bow (i.e., classes of objects to which it might refer): (a) the bow that is a physical gesture of courtesy and (b) the bow that some girls wear in their hair. Refer to them as "bow (gesture)" and "bow (hair)." By placing the following sentence:

(2) This is a curtsy.

before (1), one shifts the meaning of bow to the sense of bow (gesture). This is one observable feature of meanings:

> A. The meaning of an occurrence of a linguistic unit can be changed by changing its intra-linguistic* context.

But consider how this sort of shifting works. First, the shifting can be the effect of intra-linguistic context at arbitrarily long distances, as the sentence at the beginning of "The Brothers Karamazov" can shift the meaning of the word "brother" at the end of the book (footnote, p. 107). A second feature of the effect of intra-linguistic context is that the shifts it induces are not necessarily fixed. Thus the result of concatenating (1) and (2) does not restrict bow to the meaning of bow (gesture), as the following example illustrates:

(3) This is a curtsy. This is a bow. Tie it in your hair before you curtsy.

These shiftings give us the following features:

> A1. The meaning of an occurrence of a linguistic unit can be shifted by changing its intra-linguistic context at an arbitrary distance from that occurrence.

---

*The intra-linguistic context of an occurrence of a linguistic unit of some language L is its context with respect to occurrences of other units of that same language. The inter-linguistic context of the same occurrence of a linguistic unit is its context with respect to occurrences of units of other languages, some of which may be quite peculiar, and of types not usually dealt with by linguists (see p. 108).

A2. The shifting of meaning by means of intra-linguistic context is a process that can be iterated.

There is another class of ways in which one can narrow the meaning of the word bow in (1). Suppose that one points to the front of a boat, suggesting a third meaning which we might designate by "bow (ship)." This is another feature of meanings:

B. The meanings of an occurrence of a linguistic unit can be changed by changing the inter-linguistic context in which it occurs.

Such shifts have characteristics similar to A1 and A2.

Up to this point we have considered different meanings as if the boundaries of a meaning were fixed and all that happened to meanings of occurrences of linguistic units were that they were shifted from one territory to another by means of contexts. However, it is clear that it is no trivial matter to set the boundaries that define a single territory.

One problem is that it is not always clear when one has two distinct meanings in hand. Thus, consider the three senses of the word bow indicated by: bow (arrow), bow (violin), and bow (hair). One could argue that all three are distinct; one might argue that bow (violin) and bow (arrow) are the same, since the denotations of both seem to be constructed in similar ways; or one might argue that, since all three can be traced to the same Anglo-Saxon root, there is only one meaning involved. Thus,

C. Several meanings can be amalgamated into a single meaning.

But there is more to the matter than this. A word that may have only one meaning in one context may have many in another. Thus if one hears:

(4) Bring me a bow (instrument).

and one has available two kinds of bow (instrument) (a bow (cello) and a bow (viola)), one might well consider the request ambiguous. This sort of ambiguity appears even among those most unambiguous of linguistic units: proper names. For example, one can ask: when you say that you admire Bertrand Russell, do you mean the

"Ban the Bomb" Russell or the "Principia Mathematica" Russell? This suggests that there can be at least two senses to the proper name Bertrand Russell.

Words can have their meanings shifted, and they can also have their meanings narrowed. That is,

D. A meaning can be divided into more than one meaning.

We can shift meanings, amalgamate them, and divide them. We can also create them. The most common way in which this is done is by a special purpose device called a definition. The definition introduces a linguistic bridge from words we already understand to those we do not yet understand. In a dictionary in a language we understand such bridging to one we do not yet understand can often be done by a one-word bridge. In a dictionary within a single language, bridges often need to be somewhat longer. Still longer bridges are involved in the process called teaching.

Teaching is a translation problem that appears to be ignored by those who deal with translation. To translate the word force into the language of a sixteen-year-old who has never studied physics is a difficult process requiring bridges that may take several hours to communicate. The kinds of imagination needed for such translation suggest how difficult the problem of translating is and that one of the variables involved in the process is the person (or people) for whom the translation is intended. Thus another feature of meanings is that:

E. The total context in which an utterance appears can give it a meaning that it did not have before it appeared in that context.

Finally, it is a familiar observation that if one repeats a word often enough it ceases to have meaning:

F. The total context in which a word appears can cause the word to lose the meaning it had before it appeared in that context.

Meanings appear to have features that depend on both inter- and intra-linguistic context over arbitrarily long distances (or periods of time). The boundaries of meanings appear to be variable so that one cannot tell, in general, whether there is one meaning, many, or none.

MEANINGS AND STRING MANIPULATION

In Appendix I we represented a system's image of the environment by sets of strings and functions of strings. The features of meanings in a semantic space can now be described as operations on such sets. The axioms introduced in the section above can be restated in this light as:

A1. An input string can change the set of strings associated with a given string, no matter how long ago the latter was set up.

A2. There is no limit on how many times such a shifting of the strings associated with a given string can occur.

B. The input string can change the strings associated with a given string, even if it has a sensory modality that is different from any of the immediately associated strings.

C. Sets whose members were not associated with each other can become so.

D. Sets associated with each other can become disassociated by an input string (or better, a set of input strings).

E. Strings not associated with each other can become associated into a single set by an input string.

F. Strings which had such associated sets can lose them as the result of input strings.

We have suggested that a semantic space should have certain abstract properties. If we denote the set of strings associated (by various types of associations) with a given string A by M(A) (for the meaning of A), and we let S(. . .) be the result of adding the sentence S to a semantic representation of the form of . . . , then we can denote some* of these axioms as follows:

A1: $(\exists s)\ (\exists A)\ (S(M(A)) \neq (M(A))$

A2: $(\exists S_{i+1})\ (\exists A)\ (S_{i+1}(S_i(\ldots\ S_1(A)\ \ldots)) \neq S_i(\ldots\ S_1(A)\ \ldots))$

C: $(\exists S)\ (\exists A_1)\ (\exists A_2)\ (M(A_1) \neq M(A_2) \cdot S(M(A_1)) = S(M(A_2))$

D: $(\exists S)\ (\exists A_1)\ (\exists A_2)\ (M(A_1) = M(A_2) \cdot S(M(A_1)) \neq S(M(A_2))$

---

*We omit B.

Let I_1 and I_2 be internally stored strings:

$$E: \quad (\exists S)\,(\exists I_1)\,(\exists I_2)\,(\forall A)\,(-(I_1 \in M(A) \cdot I_2 \in M(A)) \cdot (I_1 \in S(M(A)) \cdot I_2 \in S(M(A)))$$

$$F: \quad (\exists S)\,(\exists I_2)\,(\exists I_2)\,(\exists A)\,(I_1 \in M(A) \cdot I_2 \in M(A)) \cdot -\,(I_1 \in S(M(A)) \cdot I_2 \in S(M(A)))\ .$$

An investigation of structures satisfying these axioms might have some relevance to providing mathematical machinery for handling meanings in computers, for describing the way in which such meanings are handled in both human and in man-machine systems, and for dealing with the semantics of ordinary languages.

A STRUCTURE FOR A SEMANTIC SPACE

To deal with the features of meanings discussed in the previous section, and to overcome some of the apparent weaknesses of existing theories, it may be necessary to develop a somewhat different type of mathematical structure. The current shortcomings of theories that rely heavily on meanings may merely reflect the lack of adequate (abstract) machinery for representing and manipulating meanings.

Efforts in this direction, to date, have consisted largely in attempting to apply to semantics the structures developed for other applications. There is nothing wrong with this in principle. Physics, for example, has done well using a structure developed largely to deal with the problems associated with gambling. But there may come a time to stop trying to apply the familiar and to try to develop something that is at least relatively new.

In this section we shall sketch a possible structure of the semantic space. This structure will be defined in terms of an initial set of objects, together with some rules for adding structure to this initial set. These rules will take in strings from a variety of "languages" and use these inputs to impose structure on the original set of objects. The resulting structure will then be used as the basis of other operations that might be used to answer questions or to solve problems. Two inputs interact because, when they are processed according to the rules, they influence parts of the basic structure that are close to each other within that structure, even though the inputs may be widely separated in time or space. Another way of looking at this is to consider the iterative rules as filing rules for an extensive filing system. A sentence about brothers at the beginning of a book might then influence a sentence about brothers at the end of the book because they both ended up in the same file folder.

Assume some arbitrary mathematical structures including an algebra and a topology. For example, let the algebra be the semi-group generated by the concatenation of the twenty-six letters of the English alphabet (which includes the objects A, . . . , Z , AA, . . . , ZZ , . . . , CAT, . . . , DOG, . . .) and the topology be the familiar topology of the surface of a sphere (in three-dimensional Euclidean space) with the usual metric. We shall simplify things by considering only circular neighborhoods, which we shall designate by quadruples of the form (x, y, z, w), where (x, y) indicates the coordinates of the center of the neighborhood; z, its diameter; and w, the degree of uncertainty about the exact value of (x, y).

Assume also that we have three types of links, ordered pairs of objects from the basic quadruples and/or the semi-group. Refer to the types of links as temporary, permanent, and necessary. It is these links that will be added to the structure over time to represent meanings; operations on the result will represent what is involved in handling meanings.

Consider two objects in the algebra: BLOOMINGTON and INDIANA. Select some point in the topology to represent the exact center of Indiana. An Easterner who is somewhat fuzzy about the exact locations of places in the Middle West might represent his knowledge of the location of Indiana by linking the quadruple (x′ , y′ , 300, 1000) in the topology to INDIANA in the algebra via a permanent type link. This link is supposed to represent the assumption that Indiana is a circle about 300 miles across, which he can locate accurately only within about 1000 miles of its actual position. Suppose that he now has to process the sentence:

(5) Bloomington is in Indiana.

Assume that he knows something about Bloomington (e.g., it is a city of about 20,000 inhabitants). If he has the appropriate sort of rule for generating links, he might process sentence (5) by adding a link between the quadruple (x′ , y′ , 300, 1000) and the string BLOOMINGTON.

One can now imagine how the treatment of:

(6) He is in Bloomington.

might be influenced by the result of (5), even if (6) appeared at an arbitrarily great distance from (5) in a discourse or history. This is one way in which intra-linguistic influence might be handled. If we set up a link between the written and spoken versions of the word BLOOMINGTON, sentence (5) might be in one sensory modality while (6) might be in another. This is one way in which inter-linguistic context might be handled.

Suppose that we represent the location of Illinois by linking ILLINOIS to a quadruple of the form $((x' + 150), y', 150, 500)$. (The term $(x' + 150)$ is intended to suggest that the center of Illinois is 150 miles west of the center of Indiana.) If our subject now has to process the sentence:

(7) Bloomington is in Illinois.

he may have no particular problem about there being two designations for the string BLOOMINGTON. He might link it either to the intersection of the circular neighborhoods associated with ILLINOIS and INDIANA or he might link it to their union, since in neither case does he lose or gain much information about their location. If someone asks him which direction to go from Bloomington to Boston, he can give the approximate answer "East," although we have not discussed how he might process:

(8) Which way do I go to get from Bloomington to Boston?

to get that answer.

If we postulate a third "mathematical" object that somehow represents his knowledge of the rules about airline reservations, we note that he might have difficulty in deciding whether to fly to Chicago or Indianapolis to get to Bloomington. In this case he would have considered the word BLOOMINGTON to be ambiguous, because of the other kinds of links (temporary) that he has to define to construct (the kernel of) the sentence:

(9) Get me a reservation for Indianapolis.

Context sets the meanings of words in several ways. To utter (6) adds something to the meaning of BLOOMINGTON. Uttering (5) does something different from uttering (6); this difference shows the distinction between permanent and temporary links.

RECOMMENDATIONS FOR FUTURE DEVELOPMENT

In this appendix we have suggested that the ability to describe the manner in which a command and control system handles meanings might be dealt with by a theory with a certain abstract structure, and we have roughly outlined an example of a theory that has such a structure. The machinery discussed here is not sufficiently developed to permit its direct application to the modeling of command and control systems. In order to develop this theory more fully, one might proceed in two directions. The first would study the consequences of the axioms developed in the middle of this appendix and summarized on p. 113 . The second might investigate the fuller development of the specific theory outlined on pp. 114-116.

To develop the latter kind of structure one might proceed in a number of ways. One might want to investigate rules for building up links as a function of inputs and to investigate varieties of such links. One might also want to investigate how much such structures had to have in common so that the notion of a limit of infinite sets of such structures might make sense. This limit is roughly an abstract version of the question of how much innate material there has to be for a language to be possible. (A language as a means of communication between two individuals is probably such a limit.) There will be problems of optimal coding that may shed some light on the way that the human head operates, since we can probably expect nature to be at least fairly close to optimal.

Finally, one might attempt to apply the resulting theory to the description of how meanings are handled by command and control systems. Comparing an ideal to the actual behavior of such a system in an exercise might then provide some measure of the competence of the system and some suggestions as to the directions in which improvements might be sought.

APPENDIX V

FINITE AUTOMATON MODEL*

PROBLEM

The purpose of the model described in this appendix is to provide a medium for the formal study of interactions that occur during a normative exercise.[1] The interactions that concern us are based on information transfer between the command and control system, its environment, and the exercise controllers. We are mainly concerned with two results of this interaction: (1) the training of the system and (2) the control of the command and control system's actions by the exercise controllers.

METHOD OF APPROACH

A command and control system can be thought of as a system of men and machines linked by a set of procedures. Such a system receives information from, and interacts with, its environment via a more or less extensive communications system (Figure V-1). During an exercise, this environment (or at least those of

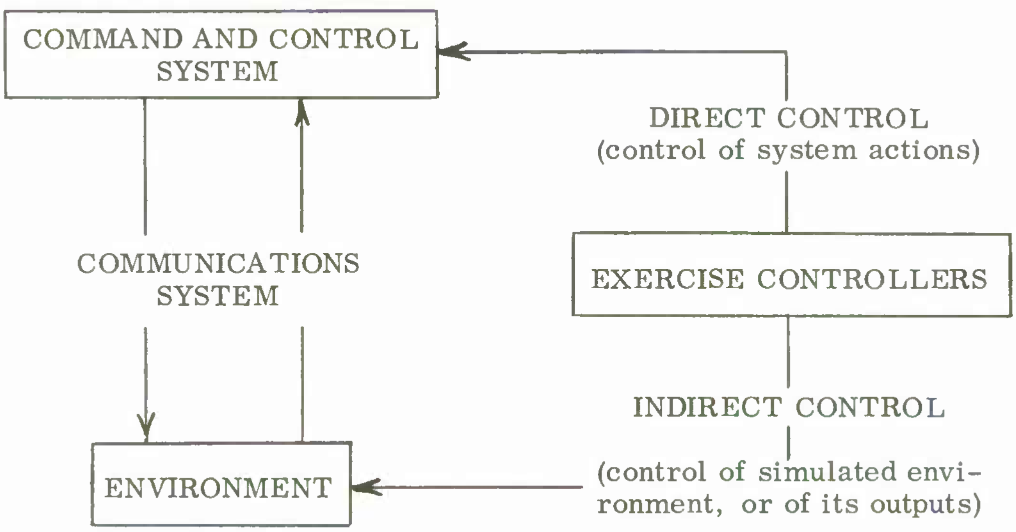


Figure V-1. Schematic Diagram of a Command and Control System in an Exercise Situation

*This model was developed by Robert A. Langevin under Technical Operations' Corporate Fellowship program.

its outputs that affect the command and control system) is simulated by the exercise controllers. They attempt to control the course of the interaction of the command and control system and the simulated environment so it proceeds logically and realistically.

We can distinguish three types of exercises: (1) open exercises, in which no control is used; (2) traditional type controlled exercises, in which control can be exercised indirectly via control over the system's inputs from its environment (Figure V-1), or directly by means of messages to the system from the controllers; and (3) normative exercises, intermediate between (1) and (2), in which control is exercised only indirectly by means of control over the simulated environmental inputs.

The model outlined here is intended as a formal system to study the interrelationship of the basic elements shown in Figure V-1 within the framework of the normative exercise. In the approach taken here, the details of the associated communications and the actual information content of the messages in the system are wholly suppressed. As a result, this model might be described as content-free. This suppression enables us to focus our attention on the interaction between the command and control system, the problem that it is attempting to solve, and the exercisers who are temporarily coupled with the system (via the problem).

This emphasis on interaction permits the introduction of the effects of training on the system in a natural way. It enables us to focus on an exercise as a learning experience for a command and control system and on the role of the exercisers as controllers of that system's behavior.

By being relatively content-free, the model gains generality. Thus, some aspects of the interactions it shows are common to a large variety of specific normative exercises. Furthermore, the model appears to formally describe a variety of situations that have a similar overall structure, e.g., various types of learning situations.

FINITE AUTOMATA

The basic elements of which the model is constructed are finite automata and simple generalizations from them. Our starting point is a paper by E. F. Moore[2] that deals with ascertaining the internal structure of a finite automaton by doing experiments on its behavior. This problem parallels that of an exerciser trying to control a command and control system in a situation where all that he can observe is its behavior. But the situation in an exercise is more complex than that described by Moore.

Finite automata are deterministic machines. Moore treats them as devices capable of having any one of a finite number of states at any given time. We will refer to these states as $q_1, \ldots, q_n$. The device is also capable of accepting any one of a finite number of input symbols $S_1, \ldots, S_m$ and of producing a finite number of output symbols $O_1, \ldots, O_p$. It reads one input symbol and produces one output symbol at any given increment of time. In general, we think of these machines as synchronous devices. Not only does time come in discrete steps or intervals, but all the conditions of the device (being in a state, reading an input symbol, and producing an output symbol) may change only from interval to interval. The state of the device at a given period of time depends only on its state in the previous increment of time and the previously read symbol. The symbol that is output by the device in a given interval depends only on the state of the machine in that interval.

Such a machine can be described by either a directed graph (transition diagram) or a table (which indicates the state at time t + 1 as a function of the symbol read and the state at time t, together with the output symbol at time t + 1 as a function of the state at t + 1). The vertexes of a directed graph represent the states of the machine and the output symbol, while the edges represent an allowable transition between states, together with the input symbol that produces that transition. Each vertex can be drawn as a circle containing the name of the state, followed by a semicolon, followed by the symbol output by the device in that state. Moore uses the illustration shown in Figure V-2.

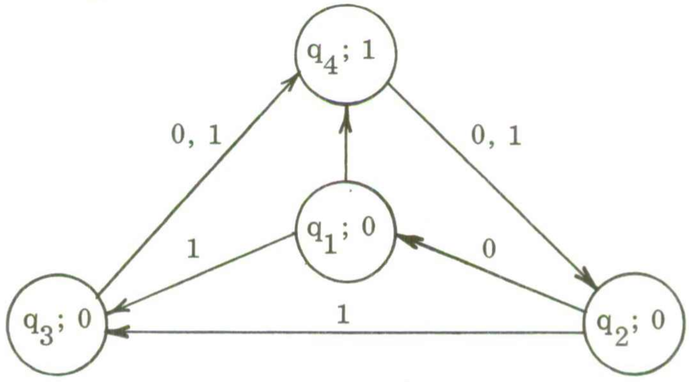


Figure V-2. Sample of a Directed Graph Representation of a Finite Automaton[2]

EXPERIMENTS WITH FINITE AUTOMATA

An experiment is performed on such a finite automaton by giving it a particular sequence of inputs. Given the machine diagrammed in Figure V-2, one might present it with the input sequence: 0001000110. If the machine is in state q_1 at the beginning of the experiment, then the machine moves through the transition sequence $q_1q_4q_2q_1q_3q_4q_2q_1q_3$ and outputs the sequence 0100010000. An experimenter with such a machine is not given the structure of the machine, he is not told its initial state, and he cannot see the internal transition sequence. He sees only the input and output sequences. Moore is concerned with what the experimenter can discover about a machine. Given a specific situation, we are concerned with how the experimenter can control the outputs of a machine by controlling its inputs.

We are concerned not simply with automata and their relationships to an experimenter, but with the interrelationships between coupled automata. The elements shown in Figure V-1 will be represented in our model by finite automata or their generalizations. The relationships of these elements will be represented as efforts of an experimenter (another finite automaton) to control a system, given only its inputs and outputs.

COMMAND AND CONTROL SYSTEMS AS COUPLED FINITE AUTOMATA

This model attempts to describe certain features of the interrelationship between these automata, which requires an extension of Moore's model. In the finite automaton that represents the command and control system, which we will henceforth call C, we single out a certain state as a terminal state. Intuitively, a sequence of outputs produced by the transitions leading to a terminal state is a solution to the exercise problem. The problem of controlling a normative exercise is to provide for a series of inputs that will lead the system to this terminal state within the time allowed for the exercise. The path taken to this state is not determined prior to the exercise, but it is constrained by the exercise plan. The path actually taken and its length are the main information that the exercisers obtain from running the exercise.

The problem of controlling the exercise is complicated by the fact that the exerciser, like the experimenter in Moore's paper, cannot see inside the system (i.e., he cannot read the minds of its personnel). One of his main problems is to exercise control without this knowledge.

Although the exerciser cannot read the minds of the personnel within C, he does have some notion of what they are thinking. He knows that system personnel have some concept of the problem that the simulated environment is presenting to them. That is, the automaton that represents the command and control system includes something that represents that system's image of the environment as presented by the exercise scenario.

The problem P is represented as a sequential machine, and the command and control system's image of this problem is represented as a sequential machine C(P) embedded into C. The command and control system's image of the problem C(P) will resemble P in some aspects but not in others.

The purpose of C is to move through the states of the problem to the terminal state. To model the checkpoints in a normative exercise, we introduce certain intermediate states that will be cut points* in the graph representing C and its environment. These are states through which the system has to pass if it is to reach the terminal state.

The exerciser (or experimenter) E will also be represented as a finite automaton. The experimenter has a copy of the problem P, and his behavior is a function of that copy. The experimenter's copy of the world will be an accurate mapping of that world, but it will also contain additional information that is used to control the behavior of C.

Thus, our model contains three basic elements: C, P, and E. Both E and C contain, as parts, copies (with possibly some errors in the case of C's copy) of P. Their "behavior" is defined as a deterministic function of the state of their copies and of their own current states. Figure V-3 shows the interrelationships between these elements; the command and control system's copy of the environment or problem is denoted by C(P), and the experimenter's copy of the environment or problem is denoted by E(P).

---

*x is a cut point of a graph G if G is connected but G is not connected if x is omitted.

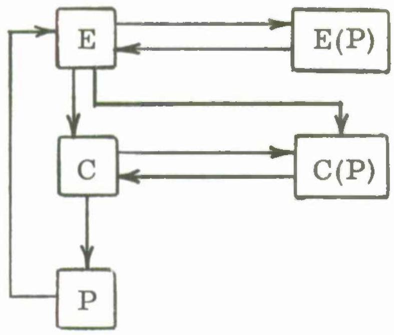


Figure V-3. Basic Parts of the Finite Automaton Model

ELEMENTS OF THE MODEL

P, the problem or environment, is a finite sequential machine composed of a set of states together with means for effecting transitions from one state to another. A state of P is a pair of integers (j, k), where j is the name of the state and k is the name of the next subgoal. (No two states have the same name.) We distinguish certain classes of states, in terms of their form alone, as follows:

(1, 1) is the initial state

(j, j) for $j < 1$ are intermediate states

(j, k) $j \neq k$ are subgoals or cut points

(j, 0) is a terminal state

We assume that j is always larger than 0.

We are concerned with machines whose inputs are taken from an alphabet of only two letters, which we denote by 0 and 1. We think of P as operating sequentially and responding, with a transition and an output, to each input that it receives. Such transitions are subject to the following conditions:

1. Any input will cause P to move from the state it is in to another unique state.

2. The state to which P moves depends only on the state of P prior to that input and the input itself.

3. After the transition induced by the input, P outputs the pair of integers which identify the state to which it has moved.

4. The goal state is a trapped state. (That is, once P reaches the goal state no further input can change its state.)

Consider a finite sequence of inputs to P, with P initially in its unique initial state. Denote this sequence of inputs by $(i_1, \ldots, i_k) = I$. We shall refer to such a sequence of inputs as "a sequence in P."

If $I = (i_1, \ldots, i_n)$ is a sequence in P and if I leads P into a terminal state, then I is called a "problem solution in P." If I is a problem solution in P such that no proper subset of I leads to the terminal state, we call n the "length of the solution." Let P be in a state S that is either initial or a subgoal, and let I be a sequence in P which takes P from the state S_j to a state S_k, which is a subgoal. If no proper subset of I leads to the terminal state, then I is called a "partial problem solution in P of length n."

We shall also make the following assumptions:

1. If S is a goal state of P there exists a problem solution in P terminating in S.

2. P contains at least one terminal state.

3. If S is a terminal state in P then there is a preferred problem solution in P terminating in S.

The command and control system's image C(P) of the problem P is not necessarily a complete or accurate picture of it. We shall represent C(P) by a graph isomorphic to the graph representing P but with different contents for the circles representing the nodes. Again, each of these circles will contain a pair of integers, but this time the second member of the pair will represent the "look-ahead" capabilities of the system if it goes to that state. Such a look-ahead capability represents the number of nodes ahead that the system can consider in determining its next transition. Thus, a node identified by the pair (4, 5) indicates the device is in state q_4, and when the system reaches that state it has a look-ahead of five nodes.

The command and control system will traverse the problem one node at a time. In effecting some transitions, it may transfer from a node, from which it can look ahead to some node N, to another node whose look-ahead is less and does not allow it to look ahead as far as node N. But it has already seen N (and used it in considering what to do). If look-ahead were interpreted in a straightforward manner, such transitions would create something corresponding to a loss of memory in the command and control system. This is clearly unrealistic.

To prevent this, we will speak of "induced look-ahead," which is a function both of the problem P and of the course of action taken by the command and control system. The notion of induced look-ahead is defined inductively. The induced look-ahead of the first node in a problem is the same as its look-ahead. If **C** goes from a node with an induced look-ahead of k to another node with a look-ahead of k', the induced look-ahead of the second node is max $(k - 1, k')$.

The exerciser E has an image of the problem E(P) that is also a graph isomorphic to that of the problem P, but where the second member of the pair within a node contains a number indicating the smallest number of nodes that must be passed to reach the terminal state.

The exerciser can aid the system in its path through the world by changing the look-ahead capabilities of various nodes in C(P). This corresponds to his providing additional information about the problem by means of inputs. He does not know the structure of C(P), but he can increase the existing look-ahead by a given amount when he finds that the number of nodes still to be traversed is incongruous with the amount of time left for problem solution. Since he does not know whether such an increment of look-ahead will actually enable the system to reach a solution, he does not control the behavior of the system. But each clue does help the system, as will become clear when we specify the behavior of C.

BEHAVIOR OF ELEMENTS

The behavior of the command and control system is determined by the following rules of procedure:

1. Don't loop (i.e., if a state has been previously reached, don't return to it).

2. If a goal or subgoal is in sight (i.e., if it is within the look-ahead capability), then go to it.

3. If a goal or subgoal is not in sight, select that transition that maximizes the (induced) look-ahead gained (i.e., that path which maximizes the sum of the amount of information gained).

4. If more than one state satisfies the conditions of (3) use some arbitrary tie breaker, either a random selection or some selection that is always used.

Assume that P is the sum of a set of graphs $P_1, \ldots, P_p$ such that each P_i has only one point in common with P_{i+1}. Each of these cut points is a subgoal. Suppose that E has available for the jth sub-problem P_j an L_j, which is the maximum desired path deviation in P_j. The behavior of E is now as follows:

From the state to which the transition has just been made, E views the sequence length (second member of the "state") from the next two possible states to which P could transfer. Let these be k_1 and k_2. E computes $|k_1 - k_2| = d$. If one of these k is 0 (i.e., if it is a subgoal) then E does nothing. Furthermore, if the deviation d is less than or equal to the maximum allowable deviation L_j, then E also does nothing. E intervenes only if $d > L_j$, in which case it modifies the second members (representing the look-ahead) of states of C(P). It does this by increasing the second member of the state by the larger k_i in C(P) and decreasing the second member of the state by the smaller k_i, both by some fixed amount. This tends to favor the shorter path. However, C selects the shorter path only if the increment added by E is sufficient to overcome the look-ahead difference that already exists in C(P).

E's modification of C(P) corresponds to the increase in information that comes from C's having made a decision and, in principle, knowing its outcome. The effect of this effort on the part of E depends on C(P), but it also depends on the size of the increments imposed by E. The decision of E, however, does not depend on any knowledge of C or C(P). However, it does change the state of C(P), and this represents learning by C. We observe the following:

1. A single intervention by E will not necessarily inhibit undesired behavior on the part of C.

2. Given sufficient intervention by E, the behavior of C will eventually be forced into the terminal state (problem solution). However, there is no guarantee that intervention of a given finite amount will necessarily accomplish this.

3. If the modified C(P) is used for the replication of a problem of the same general class (i.e., one representable by a graph with the same structure), C will be more efficient. The system thus exhibits learning or adaptive behavior with E in the role of the teacher.

4. Learning will occur without the intervention of E if we assume that look-ahead is increased at each iteration of the problem. However, learning will not be as efficient as with the intervention of E if the increment is the same in both cases.

5. Although E teaches C, one can think of E as operating with the sole motive of trying to keep the length of the actual problem solution within stated bounds. However, E need not succeed in accomplishing this aim.

COMPUTER SIMULATION

The preceding discussion specifies the finite automaton model sufficiently to make it possible to simulate it on a computer. In order to simplify the simulation we have assumed that the input and output alphabets are restricted to two letters, 0 and 1. Flow charts of such a computer simulation are shown in Figures V-4 through V-6. The data elements in these flow charts are identified in Table V-1, and the outputs are listed in Table V-2.

TABLE V-1

DATA ELEMENTS FOR FLOW CHARTS

ELEMENT	COMMENTS
State ID	Equivalent to Index
State ID ("O")	State to which "O" path leads
State ID ("1")	State to which "1" path leads
State Type (KP)	Identifies as intermediate, subgoal, or final goal
KCP	Induced look-ahead capability
ΔKCP	Intervention increment
KEP	Minimum path length to goal
K^*	Intervention limit (same throughout component)
K^* Flag	Intervention indicator
Loop Flag	Indicates entrance and exit requirements for a state
Path Flag	Indicates a state has been entered

TABLE V-2

OUTPUTS

For each State:	State ID Exit Path Rule used for choosing path Intervention occurrence indicator Use of KCP retention
For each Component:	Component ID Minimum path length Length of path followed
For each Run:	Minimum path length Length of path followed Number of interventions

Figure V-4. Flow Chart of Whole Simulation

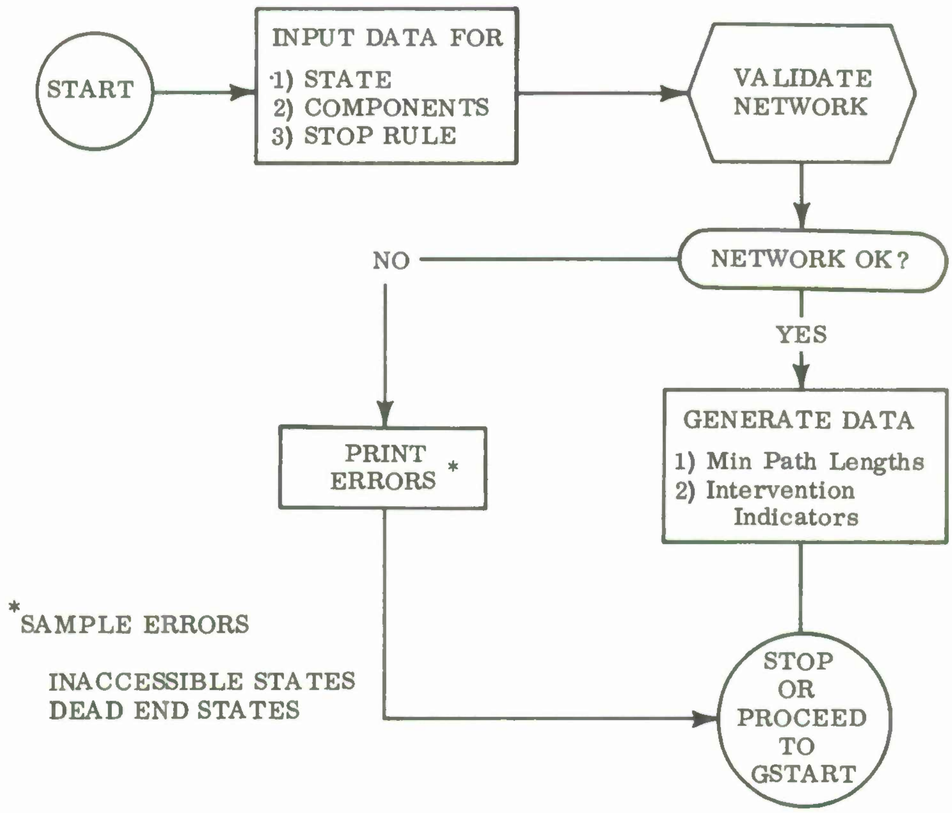


Figure V-5. Flow Chart of Preprocesser

Note: KCP = Induced Look-Ahead Capability

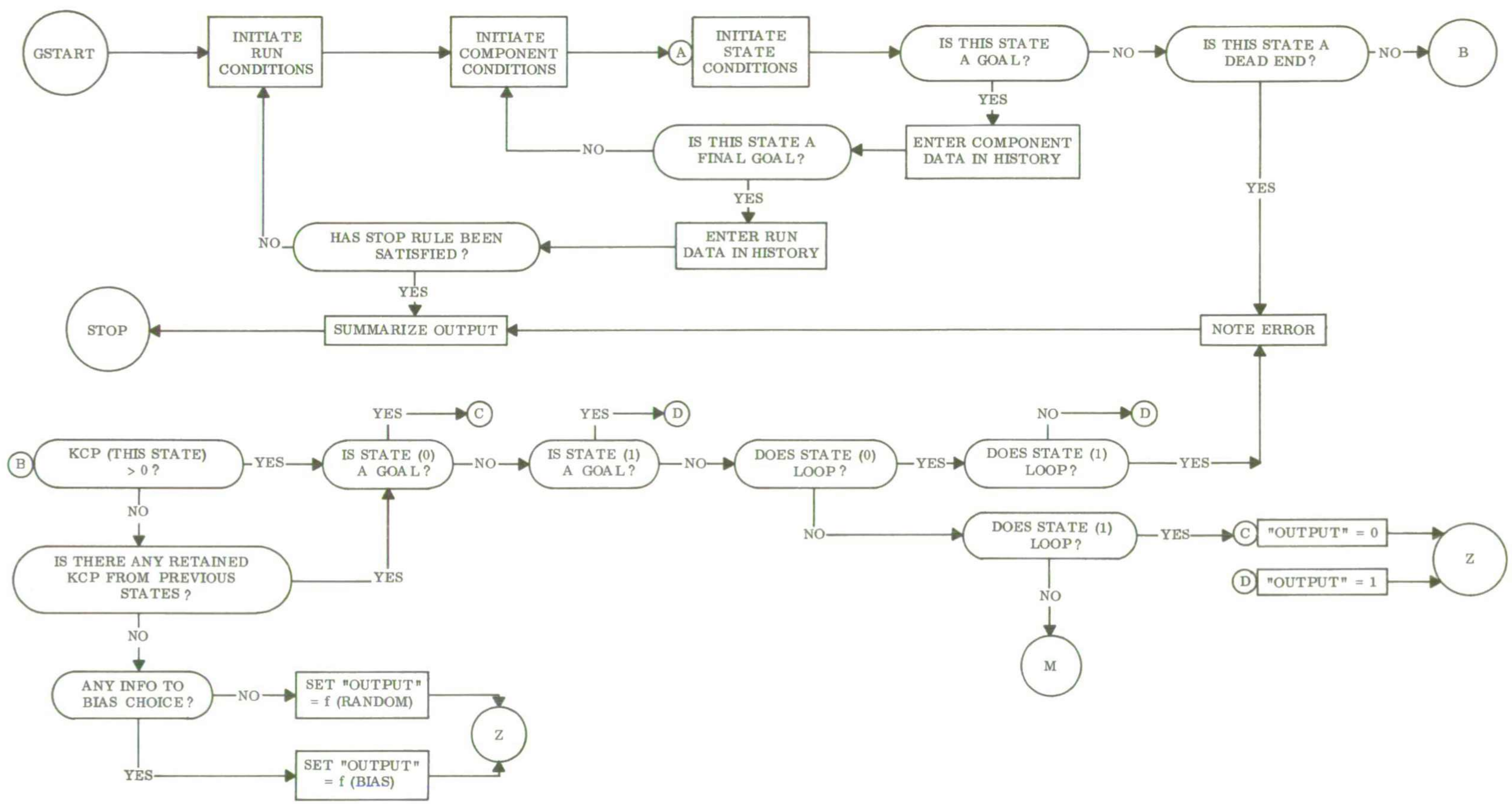


Figure V-6. Flow Chart of Simulator

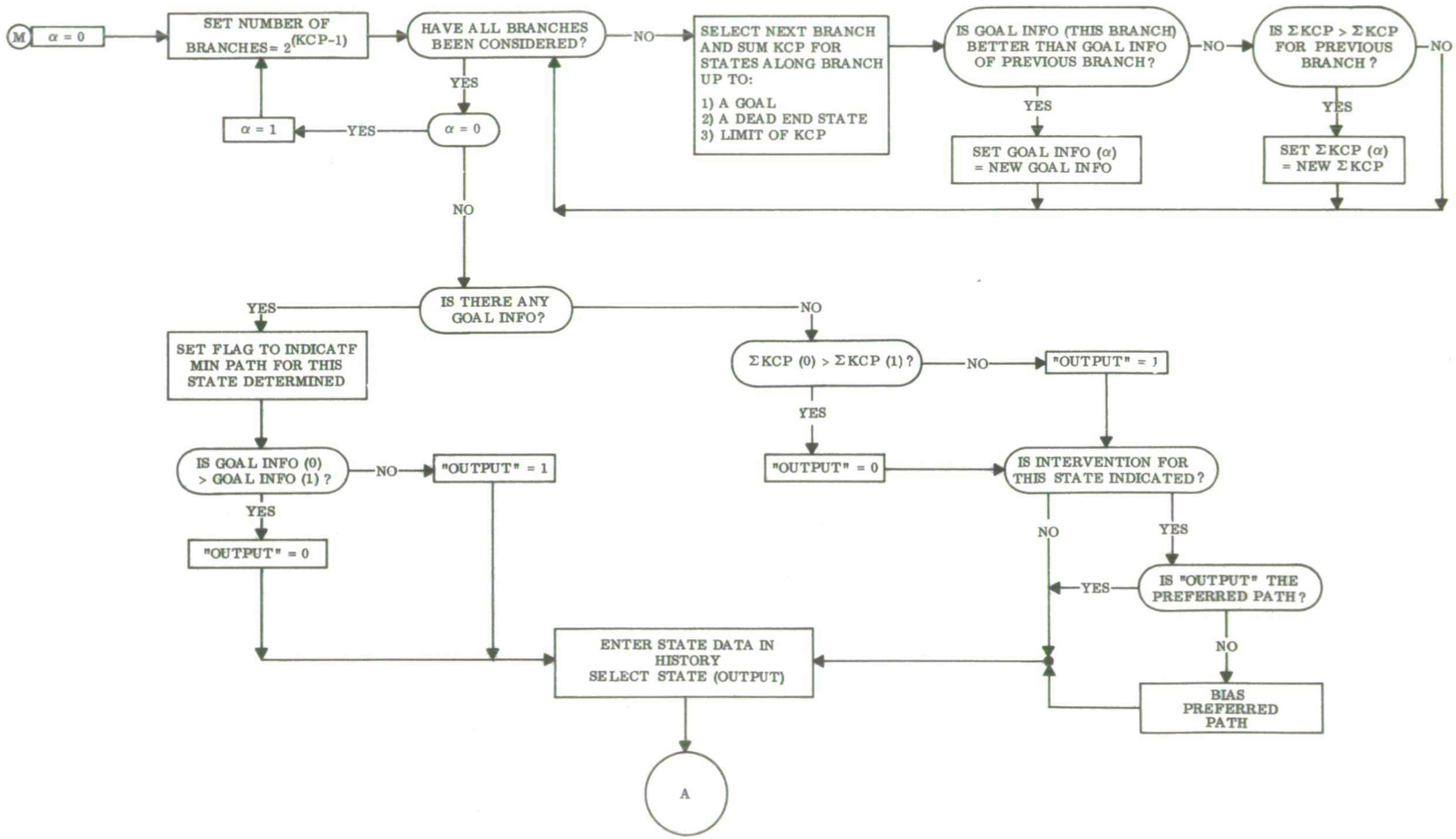


Figure V-6 (Cont'd.) Flow Chart of Simulator

TOPICS FOR FURTHER INVESTIGATION

We have shown how some of the relationships between exercisers and command and control systems in a normative exercise can be modeled by the behavior of coupled finite automata. Such a model provides a framework within which normative exercises can be studied. It also suggests the following additional problems which might be investigated:

1. What happens in this sort of situation when the automata are not perfectly reliable in their behavior (i.e., they are probabilistic automata[3] rather than deterministic automata)?

2. Are coupled automata of the deterministic kind rigorously equivalent to a single one, or does the notion of coupled automata add something new to automata theory?

3. What are the characteristics of the stimulus-response flow between the automata?

4. Under what conditions is the behavior of the coupled system convergent or divergent?

5. Suppose that the automaton C itself consists of coupled automata (i.e., personnel and machines). What is the effect of different types of coupling on the behavior of the system?

6. Can a statistical theory be developed in which large numbers of essentially identical automata variously interact?

We have shown how a computerized version of this model can be produced, and such a model might be studied empirically. Such an empirical study should suggest phenomena that would then merit theoretical investigation.

In general, this model captures various features of the interaction of systems in a "teacher-student" relation where the teacher attempts not to force a certain view on the student, but to elicit the view. A command and control system in a normative exercise is one example of this type of situation. One might want to investigate applications of this model to other situations that parallel this one, including situations in education and psychiatry.

REFERENCES

1. J. H. Proctor, "Normative Exercising: An Analytic and Evaluative Aid in System Design," IEEE Trans. on Engineering Management, E10 (1963).

2. E. F. Moore, "Gedanken-Experiments on Sequential Machines," Automata Studies (C. E. Shannon and J. McCarthy, eds.), Annals of Mathematics Studies No. 34 (Princeton, N.J.: Princeton University Press, 1956).

3. John von Neumann, "Probabilistic Logics and the Synthesis of Reliable Organisms from Unreliable Components," Automata Studies, ibid.

Unclassified

Security Classification

DOCUMENT CONTROL DATA - R&D

(Security classification of title, body of abstract and indexing annotation must be entered when the overall report is classified)

1. ORIGINATING ACTIVITY *(Corporate author)* Technical Operations Research, Burlington, Massachusetts	2a. REPORT SECURITY CLASSIFICATION Unclassified 2b. GROUP n/a

3. REPORT TITLE

Models of Command and Control Systems (With Applications to Exercise and Evaluation)

4. DESCRIPTIVE NOTES *(Type of report and inclusive dates)*

Final Report

5. AUTHOR(S) *(Last name, first name, initial)*

Kugel, Peter, and Owens, Martin F.

6. REPORT DATE	7a. TOTAL NO. OF PAGES	7b. NO. OF REFS
February, 1965		16

8a. CONTRACT OR GRANT NO. AF 19(628)-2455 b. PROJECT NO. 2801 c. TASK d.	9a. ORIGINATOR'S REPORT NUMBER(S) ESD-TDR-65-183 9b. OTHER REPORT NO(S) *(Any other numbers that may be assigned this report)*

10. AVAILABILITY/LIMITATION NOTICES

Copies have been deposited with the Defense Documentation Center. (DDC)
DDC release to CFSTI authorized.

11. SUPPLEMENTARY NOTES	12. SPONSORING MILITARY ACTIVITY
	Directorate of Computers, Electronic Systems Div., Air Force Systems Command, USAF, L.G. Hanscom Field, Bedford, Mass., 01731

13. ABSTRACT

Five models of the activities of command and control systems are described to provide a precise, if not necessarily quantitative, framework within which the behavior of command and control systems can be studied. Such models are intended as the basis for theories into which empirical data about this behavior, derived from the observation of exercises, could be fitted to provide predictions about the future behavior of particular systems. The Deductive Inference Model describes information processing as the manipulation of strings according to explicitly given rules. In terms of such a description, this model deals with the processes of problem identification and problem solving. The Inductive Inference Model deals with information processing for which the system must derive the rules that are to be used. It relates the assumptions that such a system makes and the inductive strategies that it uses to the adequacy of its predictions and generalizations. The Value Model treats a command and control system as a system that applies the values of the commander. It attempts to relate measurable features of the values held by personnel to the kinds of decisions that they make. The Semantic Model tries to deal with the manner in which command and control systems and their personnel represent their information about their environment. The Finite Automaton Model treats a command and control system and exercise controllers in certain types of controlled exercises as coupled sequential machines (finite automata). It provides a vehicle for studying the ability of the exerciser to control the behavior of the system and for studying an exercise as a learning situation.

DD FORM 1 JAN 64 1473

Unclassified

Security Classification

Unclassified

Security Classification

14. KEY WORDS	LINK A		LINK B		LINK C	
	ROLE	WT	ROLE	WT	ROLE	WT
Command and Control Systems						
System Exercising						
System Evaluation						
Mathematical Models						
Mathematical Logic						
Decision Making						
Behavior						
Analysis						
Automation						
Design Improvements						
Data Processing Systems						
Probability						

INSTRUCTIONS

1. ORIGINATING ACTIVITY: Enter the name and address of the contractor, subcontractor, grantee, Department of Defense activity or other organization *(corporate author)* issuing the report.

2*a*. REPORT SECURITY CLASSIFICATION: Enter the overall security classification of the report. Indicate whether "Restricted Data" is included. Marking is to be in accordance with appropriate security regulations.

2*b*. GROUP: Automatic downgrading is specified in DoD Directive 5200.10 and Armed Forces Industrial Manual. Enter the group number. Also, when applicable, show that optional markings have been used for Group 3 and Group 4 as authorized.

3. REPORT TITLE: Enter the complete report title in all capital letters. Titles in all cases should be unclassified. If a meaningful title cannot be selected without classification, show title classification in all capitals in parenthesis immediately following the title.

4. DESCRIPTIVE NOTES: If appropriate, enter the type of report, e.g., interim, progress, summary, annual, or final. Give the inclusive dates when a specific reporting period is covered.

5. AUTHOR(S): Enter the name(s) of author(s) as shown on or in the report. Enter last name, first name, middle initial. If military, show rank and branch of service. The name of the principal author is an absolute minimum requirement.

6. REPORT DATE: Enter the date of the report as day, month, year, or month, year. If more than one date appears on the report, use date of publication.

7*a*. TOTAL NUMBER OF PAGES: The total page count should follow normal pagination procedures, i.e., enter the number of pages containing information.

7*b*. NUMBER OF REFERENCES: Enter the total number of references cited in the report.

8*a*. CONTRACT OR GRANT NUMBER: If appropriate, enter the applicable number of the contract or grant under which the report was written.

8*b*, 8*c*, & 8*d*. PROJECT NUMBER: Enter the appropriate military department identification, such as project number, subproject number, system numbers, task number, etc.

9*a*. ORIGINATOR'S REPORT NUMBER(S): Enter the official report number by which the document will be identified and controlled by the originating activity. This number must be unique to this report.

9*b*. OTHER REPORT NUMBER(S): If the report has been assigned any other report numbers *(either by the originator or by the sponsor)*, also enter this number(s).

10. AVAILABILITY/LIMITATION NOTICES: Enter any limitations on further dissemination of the report, other than those imposed by security classification, using standard statements such as:

(1) "Qualified requesters may obtain copies of this report from DDC."

(2) "Foreign announcement and dissemination of this report by DDC is not authorized."

(3) "U. S. Government agencies may obtain copies of this report directly from DDC. Other qualified DDC users shall request through

___________________________________."

(4) "U. S. military agencies may obtain copies of this report directly from DDC. Other qualified users shall request through

___________________________________."

(5) "All distribution of this report is controlled. Qualified DDC users shall request through

___________________________________."

If the report has been furnished to the Office of Technical Services, Department of Commerce, for sale to the public, indicate this fact and enter the price, if known.

11. SUPPLEMENTARY NOTES: Use for additional explanatory notes.

12. SPONSORING MILITARY ACTIVITY: Enter the name of the departmental project office or laboratory sponsoring *(paying for)* the research and development. Include address.

13. ABSTRACT: Enter an abstract giving a brief and factual summary of the document indicative of the report, even though it may also appear elsewhere in the body of the technical report. If additional space is required, a continuation sheet shall be attached.

It is highly desirable that the abstract of classified reports be unclassified. Each paragraph of the abstract shall end with an indication of the military security classification of the information in the paragraph, represented as *(TS), (S), (C), or (U)*.

There is no limitation on the length of the abstract. However, the suggested length is from 150 to 225 words.

14. KEY WORDS: Key words are technically meaningful terms or short phrases that characterize a report and may be used as index entries for cataloging the report. Key words must be selected so that no security classification is required. Identifiers, such as equipment model designation, trade name, military project code name, geographic location, may be used as key words but will be followed by an indication of technical context. The assignment of links, rules, and weights is optional.

File With Report

EDITOR'S CHECKLIST FOR TECHNICAL REPORT FORMAT

ESD TDR NUMBER
65-183 (ESRCC) Tech. Ops Research (Final Report)

DATE
10 Mar 65 (Cont. 19(628)-2455

FRONT COVER

- [ ] NO DISCREPANCIES
- [ ] COVER OMITTED
- [ ] CLASSIFICATION MARKING OMITTED
- [ ] ESD IDENTIFICATION OMITTED
- [ ] IMPROPER COVER STOCK
- [x] IMPROPER FORMAT (please use attached mock-up)
- [ ] DATE OMITTED
- [ ] OTHER

SPECIAL NOTICES (Inside Front Cover)

- [ ] NO DISCREPANCIES
- [x] ALL SPECIAL NOTICES OMITTED (now included on mock-up cover.)
- [ ] AVAILABILITY NOTICE OMITTED
- [ ] DISSEMINATION NOTICE OMITTED
- [ ] DISPOSITION NOTICE OMITTED
- [ ] LEGAL NOTICE OMITTED
- [ ] OTHER

TITLE PAGE (First right-hand page)

- [ ] NO DISCREPANCIES
- [ ] AUTOMATIC DOWNGRADING NOTICE OMITTED
- [ ] ESPIONAGE NOTICE OMITTED
- [ ] CLASSIFICATION NOTICE OMITTED
- [ ] TITLE PAGE OMITTED
- [ ] TITLE OMITTED
- [x] OTHER (included, but improper format)

Foreword ~~FOREWORD (Reverse of Title page)~~

- [ ] NO DISCREPANCIES
- [x] FOREWORD PAGE OMITTED (see item # 1, reverse)
- [ ] SYSTEM, PROJECT OR TASK NUMBER OMITTED
- [ ] CONTRACTOR'S NAME AND ADDRESS OMITTED
- [ ] DISTRIBUTION LIMITATION REASON OMITTED
- [ ] CONTRACT NUMBER OMITTED
- [ ] DATE OMITTED
- [ ] CAPTION OMITTED
- [ ] OTHER

APPROVAL STATEMENT (~~Last page of Foreword~~)

- [ ] NO DISCREPANCIES
- [x] APPROVAL STATEMENT OMITTED (see item # 2, reverse)
- [ ] OTHER

ABSTRACT (Next right-hand page)

- [x] NO DISCREPANCIES
- [ ] ABSTRACT OMITTED
- [ ] EXCESSIVE LENGTH
- [ ] OTHER

(Continued on reverse side)

TEXT

- [ ] NO DISCREPANCIES
- [ ] NOT DIVIDED INTO SECTIONS
- [ ] CLASSIFICATION MARKING OMITTED
- [ ] PAGE NUMBERING OMITTED
- [ ] PAGE NUMBERS IMPROPERLY PLACED
- [ ] PAGES PRINTED ONE SIDE ONLY
- [ ] DOUBLE SPACING
- [ ] IMPROPER FOLDOUTS
- [X] OTHER (see item #3, below)

DD FORM 1473 (Last printed page)

- [ ] NO DISCREPANCIES
- [X] FORM 1473 OMITTED (see item #4, below)
- [ ] INCOMPLETE
- [ ] OTHER

BACK COVER

- [X] NO DISCREPANCIES
- [ ] COVER OMITTED
- [ ] CLASSIFICATION MARKINGS OMITTED
- [ ] OTHER

BINDING

- [X] NO DISCREPANCIES (see item #5, below)
- [ ] IMPROPERLY BOUND
- [ ] OTHER

ADDITIONAL REMARKS

1. Foreword. A foreword is not required but is recommended. It may include names of authors, credit for assistance received, contractor report number or any additional information desired.

2. Review and Approval. A review and approval statement signed by the Air Force Project Officer must be included in the report. This statement should appear at the bottom of the abstract page, space permitting, or the page following. (ref par 19, ESD Contractor Reports Exhibit 63-1, Vol III, May 1964.)

3. Text. The text set up is very good, the space and one half image is acceptable, and it is assumed by the way the pages are numbered that final copy will be back-to-back. However, there is one item that is not acceptable and that is the blue pages that are included; we request that they be omitted on final copy and that all pages be printed black characters on white opaque paper. ESD Technical Reports are subject to further re-production by the Defense Documentation Center (DDC) and other government agencies by microfilm process and other black and white methods in order to satisfy needs for additional quantities; this should be borne in mind when final copy is prepared.

4. DD Form 1473. This item is not required under Vol. III, ESD Contractor Reports Exhibit 63-1. I have included a partially completed form; please complete Item # 9.b., if applicable, and include one copy of the form as the last page in each copy of the report.

(see more on attached sheet)

10 Mar 65

ESRCC (Major Mellin)

Please include the review and approval statement on page following abstract. The statement should read as follows: "This technical documentary report has been reviewed and is approved" and should be signed by the project officer.

If we may assist you further, please contact us.

Wreck